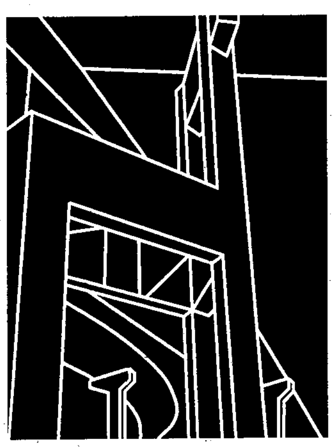
RESEARCH REPORT 987-6

TRAFFIC-LOAD FORECASTING USING WEIGH-IN-MOTION DATA

Tongbin Qu, Clyde E. Lee, and Liren Huang



CENTER FOR TRANSPORTATION RESEARCH
BUREAU OF ENGINEERING RESEARCH
THE UNIVERSITY OF TEXAS AT AUSTIN

MARCH 1997

Technical Report Documentation Page

1. Report No. TX-99/987-6	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle TRAFFIC-LOAD FORECASTING USING WEIGH-IN-MOTION DATA		5. Report Date March 1997
7. Author(s) Tongbin Qu, Clyde E. Lee, and Liren Huang		6. Performing Organization Code 8. Performing Organization Report No. 987-6
9. Performing Organization Name and Address Center for Transportation Research The University of Texas at Austin 3208 Red River, Suite 200 Austin, TX 78705-2650		10. Work Unit No. (TRAIS) 11. Contract or Grant No. 7-987
12. Sponsoring Agency Name and Address Texas Department of Transportation Research and Technology Transfer Section/Construction Division P.O. Box 5080 Austin, TX 78763-5080		13. Type of Report and Period Covered Research Report (9/95-8/96) 14. Sponsoring Agency Code

15. Supplementary Notes
Project conducted in cooperation with the Federal Highway Administration.

16. Abstract

Vehicular traffic loading is a crucial consideration for the design and maintenance of pavements. With the help of weigh-in-motion (WIM) systems, the information about date, time, speed, lane of travel, lateral lane position, axle spacing, and wheel load for each vehicle passing a WIM site can be recorded continuously on-site and transferred to a remote computer. This study focused on using the data from two WIM systems installed to support research on pavement performance.

Data analysis involved processing the data from the two WIM systems, summarizing the data into respective TxDOT vehicle classes, analyzing the error records in the WIM data, and exploring trends and patterns in the observed traffic counts. The researchers also analyzed the axle-load frequency distribution for different axle groups within all truck classes, explored the trend of axle-load distribution among years, and compared the difference in axle-load distribution for the same axle type at different locations in different vehicle classes. A time-series method was used to forecast traffic counts for each vehicle class based on the trend in the pattern of observed traffic and a growth rate for each vehicle class. Finally, cumulative traffic loading was forecasted by applying the estimates of future traffic count to the respective axle-load frequency distribution. A C-language computer program that runs on PC-compatible machines was developed to facilitate data processing for traffic-load forecasting.

17. Key Words Weigh-in-motion systems, vehicle classification, axle weight frequency distribution, weighted average ESALs per vehicle, traffic count forecasting, traffic load forecasting	18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.		
19. Security Classif. (of report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 132	22. Price

Form DOT F 1700.7 (8-72) Reproduction of completed page authorize

TRAFFIC-LOAD FORECASTING USING WEIGH-IN-MOTION DATA

by

Tongbin Qu,

Clyde E. Lee,

and

Liren Huang

Research Report Number 987-6

Research Project 7-987

A Long-Range Plan for the Rehabilitation of US 59 in the Lufkin District

conducted for the

TEXAS DEPARTMENT OF TRANSPORTATION

by the

CENTER FOR TRANSPORTATION RESEARCH
Bureau of Engineering Research
THE UNIVERSITY OF TEXAS AT AUSTIN

March 1997

IMPLEMENTATION RECOMMENDATION

The computer program developed for this study of vehicle sorting, axle-load frequency distribution, and future traffic load forecasting can be used on any PC computer; however, it was implemented on a Pentium computer. The program is intended to facilitate the analysis of observed traffic loading and the forecasting of future traffic loading. It is valid for WIM systems that use the same data format as the one from which the research data was obtained (PAT DAW100).

Prepared in cooperation with the Texas Department of Transportation

DISCLAIMERS

ACKNOWLEDGMENTS

This study is part of the continuing research study on the long-range pavement rehabilitation plan for the portion of US 59 within the Lufkin District. Sincere appreciation is expressed to Eric Starnater, Lufkin District Pavement Engineer, who has assisted this research study with WIM data collection and trouble shooting of the WIM systems, and to Liren Huang, research assistant at the Center for Transportation Research at The University of Texas at Austin, who gave his valuable suggestions and support on data processing and analysis in operating the WIM systems.

Clyde E. Lee, P.E. (Texas No. 20512)
Research Supervisor

TABLE OF CONTENTS

SUMMARY

Vehicular traffic loading is a crucial consideration for the design and maintenance of pavements. With the help of weigh-in-motion (WIM) systems, the information about date, time, speed, lane of travel, lateral lane position, axle spacing, and wheel load for each vehicle passing a WIM site can be recorded continuously on-site and transferred to a remote computer. This study focused on using the data from two WIM systems installed to support research on pavement performance.

Data analysis involved processing the data from the two WIM systems, summarizing the data into respective TxDOT vehicle classes, analyzing the error records in the WIM data, and exploring trends and patterns in the observed traffic counts. The researchers also analyzed the axle-load frequency distribution for different axle groups within all truck classes, explored the trend of axle-load distribution among years, and compared the difference in axle-load distribution for the same axle type at different locations in different vehicle classes. A time-series method was used to forecast traffic counts for each vehicle class based on the trend in the pattern of observed traffic and a growth rate for each vehicle class. Finally, cumulative traffic loading was forecasted by applying the estimates of future traffic count to the respective axle-load frequency distribution. A C-language computer program that runs on PC-compatible machines was developed to facilitate data processing for traffic-load forecasting.

CHAPTER 1. INTRODUCTION

1.1 GENERAL

Vehicular traffic loading is a crucial consideration for the design and maintenance of pavements. Truck traffic has been constantly on the rise for the past half century. As a consequence, highway engineers continue to face the challenge of designing and maintaining pavements to accommodate increasing truck traffic loading. Perhaps the most significant problem faced by highway design engineers is the non-availability of timely, reliable traffic data, such as truck traffic percentage and axle loads. Pavement damage is a direct function of axle loads; therefore, only reliable and accurate traffic data can produce good pavement design and maintenance solutions.

With the advent of weigh-in-motion (WIM) technology, the collection of large, representative samples of traffic load data has become more efficient and effective. Besides axle load information, a WIM system can also obtain information about speed, lane of operation, date and time of vehicle passage, and the number and spacing of axles. Furthermore, the average daily traffic (ADT) for a certain period of time, or the average annual daily traffic (AADT), can be calculated directly when a WIM system continuously counts and records all vehicles that pass through a WIM site.

WIM data can also be used for forecasting future traffic loading. Two key traffic factors — the future truck volume and the corresponding axle weight frequency distributions — can be estimated after exploring the growth pattern of the observed truck volume and axle weight frequency distributions obtained from WIM data.

1.2 BACKGROUND

US 59 is a principal arterial highway that runs from Laredo through Houston and Lufkin, exits Texas at Texarkana, and extends northeast all the way to Canada. The section of this road in Texas carries a considerable amount of truck traffic. The Texas Department of Transportation (TxDOT), in cooperation with the Center for Transportation Research (CTR) of The University of Texas at Austin, initiated a research project to develop a long-range rehabilitation plan for US 59 in the Lufkin District (about 225 km centerline length). Two WIM systems, augmented with infrared detectors and temperature sensors, were installed in two pavement test sections for the purpose of continuously collecting traffic and temperature data. Both WIM systems are in the southbound traffic lanes of US 59 and are located, respectively, to the north (rigid pavement) and to the south (flexible pavement) of Corrigan, about 10 km apart.

1.3 OBJECTIVES

The overall objective of this portion of the study is to develop a procedure for using WIM system data to estimate future traffic volume and axle loads, and to calculate the probable pavement damage resulting from such loads in terms of equivalent 18-kip equivalent single axle loads (ESALs). This objective will be achieved through the following steps:

- Analysis of observed traffic: Develop a new computer program to substitute for the previous Excel macro program. The new program will sort vehicles by TxDOT's current classification scheme (number and spacing of axles) and will be time efficient.
- Traffic forecasting: Analyze the pattern of observed traffic volumes by vehicle class and develop an annual growth rate for each vehicle class.
- Axle weight frequency distribution: Analyze the observed axle weight frequency distributions by vehicle classes and explore their individual growth trends.

Since truck traffic usually contributes a major proportion of all ESALs (Ref 18), this study emphasizes the effects of truck traffic, especially the 5-axle single trailer, which comprised a large percentage of the observed trucks.

1.4 ORGANIZATION OF THE REPORT

A brief description of the project background and the proposed objectives has been presented in Chapter 1. The computer program that was developed for sorting vehicles by the current TxDOT classification scheme and for analyzing the observed traffic patterns is introduced in Chapter 2.

Analyses of the patterns of observed vehicle traffic counts and a means for selecting an appropriate growth rate for the next 20 years are presented in Chapter 3. Comparisons of the axle weight frequency distributions of different truck classes and patterns, as well as trends of these distributions, are presented in Chapter 4.

A concept developed at the American Association of State Highway Officials (AASHO) Road Test for relating traffic loads to pavement damage, namely, the idea of the equivalent single axle load (ESAL), is applied to observed WIM data and an example of forecasted ESALs is presented in Chapter 5. Finally, conclusions drawn from this research and recommendations for further study are presented in Chapter 6.

CHAPTER 2. DATA PROCESSING AND ANALYSIS OF OBSERVED TRAFFIC

The first step in estimating future traffic loads is to process the observed weigh-in-motion (WIM) data. Efficient processing of data at the first stage is essential, as large digital data files are created by the WIM system and are stored temporarily on site. After being downloaded to a remote computer, the data in these files must be formatted suitably and used to extract information about traffic characteristics with respect to time. The product of the data processing effort made for processing data reported herein is a computer program designed to efficiently provide traffic and weight information from observed WIM data.

2.1 WEIGH-IN-MOTION (WIM) SYSTEM

A WIM system, as its name implies, weighs vehicles in motion. Historically, vehicle weight data have been collected by stopping vehicles at weigh stations and weighing each axle of the static vehicle on scales. This procedure was time consuming, expensive, and often hazardous, and only small samples could be obtained.

The WIM concept, which involves the measurement and analysis of dynamic vehicle tire forces for estimating the corresponding gross-vehicle weight, and the portion thereof carried by each wheel or axle of a moving vehicle, offers many advantages over conventional static weighing. A WIM system can weigh and dimension virtually every vehicle that passes, and the data can be accessed from remote locations. Thus, a WIM system has a relatively low operating cost per vehicle weighed. Although a WIM system might have a higher initial hardware cost, compared with scales used for static weighing, it is much more cost effective for obtaining representative samples of statistical data related to traffic loading characteristics.

2.1.1 Components of a WIM System

The WIM system used for this project comprises four basic components:

- Tire force sensors (weigh pads)
- Vehicle presence detector (inductance loop)
- Tire sensor (infrared light beam)
- Signal processor unit

For details of each component and its function, readers are encouraged to consult the *System Description of Automatic Vehicle Weight and Classification System* by PAT Equipment Corporation, Inc. (now PAT Traffic Control Corporation) (Ref 19). Figure 2.1 shows the layout of the WIM system.

2.1.2 Location

Two WIM systems monitored southbound US 59 traffic north and south of the US 287–US 59 intersection at Corrigan. Figure 2.2 shows the geographic location of the WIM systems.

2.2 TEXAS DEPARTMENT OF TRANSPORTATION (TXDOT) CLASSIFICATION

Various vehicle classification schemes have been used over the years. TxDOT revised and published the vehicle classification currently used by their observers and analysts in January 1996 (Ref 20). Vehicles are classified into thirteen basic types. Each type may have one or more chassis or axle-spacing configurations. The axle-spacing range for TxDOT classification is shown in Table 2.1, and typical vehicle profiles for each type are sketched in Figure 2.3.

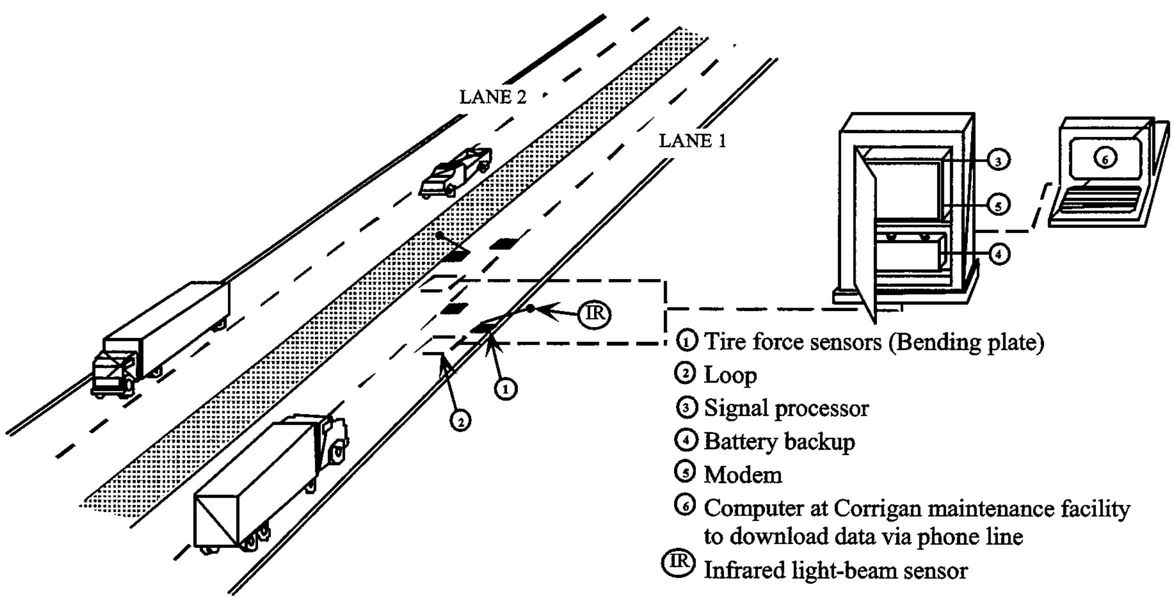


Figure 2.1 Layout of WIM system

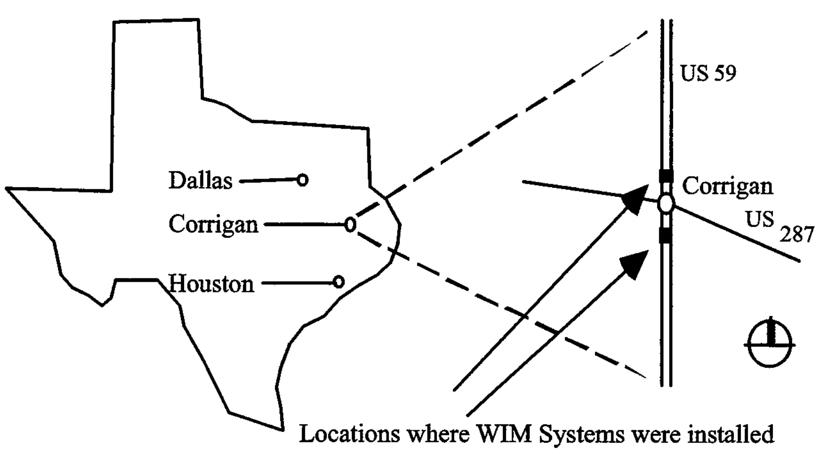


Figure 2.2 Geographic location of WIM systems

Table 2.1 TxDOT vehicle classification table (by axle spacing)

TYPE	CLASS	A - B	B - C	C - D	D - E	E - F	F - G
1	MTR. CYCLE - CAR	0.1 - 10.2					
1	CAR. 1 AXLE TR.	6.1 - 10.2	6.0 - 20.1				
1	CAR. 2 AXLE TR.	6.1 - 10.2	6.0 - 20.1	0.1 - 3.3			
2	PICK-UP	10.3 - 13.0					
2	PICK-UP -1AX TR.	10.3 - 13.0	6.0 - 20.1				
2	PICK-UP -2AX TR.	10.3 - 13.1	6.0 - 20.1	0.1 - 3.3			
3	BUS - 2 AXLE	21.0 - 40.0					
3	BUS - 3 AXLE	21.0 - 40.0	3.4 - 6.0				
4	2 D	13.1 - 20.9					
4	2 D - 1 AXLE-TR.	13.1 - 20.9	6.1 - 20.1				
4	2 D - 2 AXLE-TR.	13.1 - 20.9	6.1 - 20.1	0.1 - 3.3			
5	3 AX. SINGLE UN (3A)	6.1 - 20.9	3.4 - 4.7				
6	4 AX. SINGLE UN (4A)	13.1-20.9	3.4 - 4.7	3.4 - 4.7			
6	4 AX. SINGLE UN (RIG)	0.1 - 6.0	13.1 - 29.0	3.4 - 6.0			
7	2S1	6.1 - 20.0	20.2 - 60.0				
8	2S2	6.1 - 20.0	16.5 - 40.0	3.4 - 6.0			
8	3S1	6.1 - 20.0	3.4 - 6.0	6.1 - 40.0			
9	2S3	6.1 - 25.0	6.1 - 40.0	3.4 - 6.0	3.4 - 6.0		
9	3S2	6.1 - 25.0	3.4 - 6.0	6.1 - 40.0	3.4 - 12.0		
10	3S3 (SINGLE TR.)	6.1 - 22.0	3.4 - 6.0	10.4 - 40.0	3.4 - 6.0	3.4 - 6.0	
10	3S4 (SINGLE TR.)	6.1 - 22.0	3.4 - 6.0	10.4 - 40.0	3.4 - 6.0	3.4 - 6.0	3.4 - 6.0
11	2S1-2 (DBL. TR.)	6.1 - 17.0	11.1 - 23.0	6.1 - 18.0	11.1 - 23.0		
12	2S2-2 (DBL. TR.)	6.1 - 17.0	11.1 - 23.0	3.4 - 6.0	6.1 - 18.0	11.1 - 23.0	
12	3S1-2 (DBL. TR.)	6.1 - 25.0	3.4 - 6.0	6.1 - 40.0	6.1 - 18.0	11.1 - 23.0	
13	3S2-2	6.1 - 17.0	3.4 - 6.0	11.1 - 23.0	3.4 - 6.0	6.1 - 18.0	11.1 - 23.0
14	UNCLASSIFIED						

2.3 SORTING WIM DATA BY VEHICLE TYPE

Data files transferred from the PAT DAW 100 system are in binary code. These data can be converted to ASCII code by a program — dubbed WIMFTP — developed by Liren Huang at the Center for Transportation Research of The University of Texas at Austin. The computer program developed by the authors for this study operates on the data files after they are converted to ASCII code.

2.3.1 Data Format of the ASCII Data File

- File name format
  Each daily data file is stored in the PAT DAW 100 system under a unique file name using the "Dsssmmdd.yy" format, where:
    D: Raw data file designator
    sss: Site number, (i.e., 001 for site 1)
    mm: Month
    dd: Day
    .: Extension separator
    yy: Year
    After the data file is converted to ASCII code, the file name is changed from "Dssmdd.yy" to "Vsssmmdd.yy".

- ASCII data file format
  An ASCII data file is composed of strings. Each vehicle generates a string of data when it passes through the WIM system. The following is an example of a string drawn from file V0020324.94 (March 24th of 1994, site 2):
  2,3,24,0,0,26,63,28,12,1.2,1.2,12,0.8,0.7,9.4
  where:
    2: Lane used by vehicle (i.e., 1 for right lane, 2 for left lane)
    3:Month
    24: Day
    0: Hour
    0: Minute
    26: Second
    63: Speed of first axle (mi/h)
    28: Time, used to calculate lateral position (ms)
    12: Infrared blocked time for the first axle (ms)
    1.2: Load of the left wheel of the first axle (kip)
    1.2: Load of the right wheel of the first axle (kip)
    12: Infrared blocked time for the second axle (ms)
    0.8: Load of the left wheel of the second axle (kip)
    0.3: Load of the right wheel of the second axle (kip)
    9.4: Axle spacing between the first and second axle (ft)

The above data string resulted from a two-axle vehicle. Vehicles with more than two axles have four additional numbers (separated by commas) in the string for each additional axle. The order of the four succeeding numbers is in the sequence: infrared blocked time for the axle, left wheel load for the axle, right wheel load for the axle, axle spacing between the previous axle and the current axle. Hence, the string length of a two-axle vehicle contains 15 data points; the string length of a three-axle vehicle contains 19 data points, and so on.

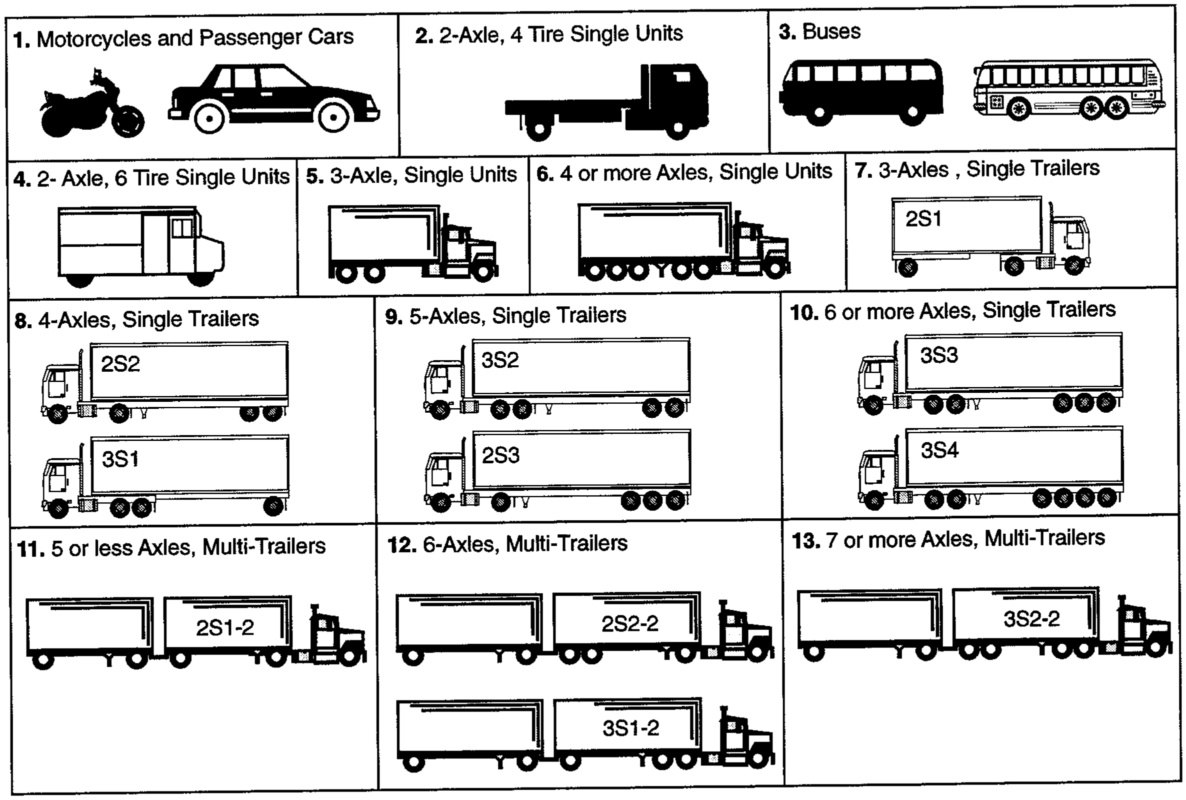


Figure 2.3 Typical vehicle profiles for TxDOT vehicle type

2.3.2 The Algorithm of the Program

The computer program that was developed for sorting data into TxDOT vehicle classes is written in C language. The basic logic of the program is to determine the number of axles on a vehicle by calculating the length of the data string and then categorizing the vehicle as one of thirteen vehicle types (or unclassified) according to TxDOT's axle-spacing criteria (see Table 2.1).

Some adjustments were made to TxDOT's range of axle spacing, since a number of vehicles contained in the data file had an axle spacing pattern that was only slightly different from the standard axle spacings in the table. The first adjustment made was to expand the last axle spacing of vehicles in Type 12 (2S2-2 & 3S1-2) from 11.1–23.0 ft to 11.1–25.0 ft.* The second adjustment was to reduce the lower limit of the tandem axle spacing of the vehicles in Type 9 (3S2 & 2S3) from 3.4 ft to 2.5 ft. It was often observed in the error records that the last axle spacing recorded for a Type 12 truck was slightly over 23.0 ft and that the recorded tandem-axle spacing of a few 5-axle single trailer trucks was somewhat less

* Because TxDOT's axle-spacing criteria are given in Imperial units, we use similar units, rather than SI units, in this report.

than 3.4 ft. After these two adjustments were made, total daily error records were reduced to about 0.5 percent, as error records were usually fewer than 50 vehicles per day.

The infrared sensors were inoperative for various reasons and at various times during the 3-year period. When an infrared sensor was not functioning, the infrared blocked time in the ASCII data file was unreasonably large and started with a "%" sign. In order to use the satisfactory data relating to axle load and spacing, the program ignores any infrared blocked time value that starts with the "%" sign.

Powered by a Pentium PC computer, the program can process 30 days of WIM data (about 7,000 vehicles per day) in less than 10 minutes. The flow chart of the program is shown in Figure 2.4, while the code for the program is provided in Appendix A. Appendix B includes tables showing the observed vehicles sorted into thirteen types and error records (which are stored as Type 14) for each month of 1995.

2.3.3 Error File Analysis

After sorting the data by vehicle class, the record for any vehicle whose axle spacing does not match that of any listed vehicle class is designated an error record. The program can count the number of error records and display them upon request. There are several causes for error records. The most common are:

Off scale — When a vehicle changes lanes or one side of a vehicle is driving on the shoulder at the WIM site, the wheels on one end of the axle do not contact either weigh pad in the lane. The wheels are, therefore, off scale and a zero wheel weight is registered by the WIM system. Once the weight of one side of an axle is missing, a large and improbable axle spacing following the zero weight is recorded by the system.

Several vehicles in one record — Analysis of some of the error records indicates that some of the unusually long records are actually a combination of several vehicles, e.g., a 5-axle single trailer combined with a passenger car, or several passenger cars combined in the same record. This type of error might have resulted from using an inappropriate value for the extension time of the inductance loop on the DAW 100. A long loop extension time can cause the DAW 100 to combine successive vehicles into a single vehicle record.

Unreasonable axle spacing — Some records have unreasonable axle spacing, but they are not associated with off-scale conditions. The cause for such records has not been determined.

Error in axle spacing compared with TxDOT classification — Even though some axle spacing ranges in the current TxDOT classification scheme have been increased to accommodate some observed non-standard vehicles in the computer program, there are still a few data records that show vehicles of a slightly different axle arrangement. Since these error records comprise a low percentage of all error records (about 5 percent), they are eliminated from further analysis.

Ghost record — Ghost records refer to "the occurrence of records that were generated without the presence of any vehicles" (Ref 18). This kind of error record appeared frequently in the data files during 1993. However, the WIM system manufacturer adjusted the system software and such records do not appear in the more recent data.

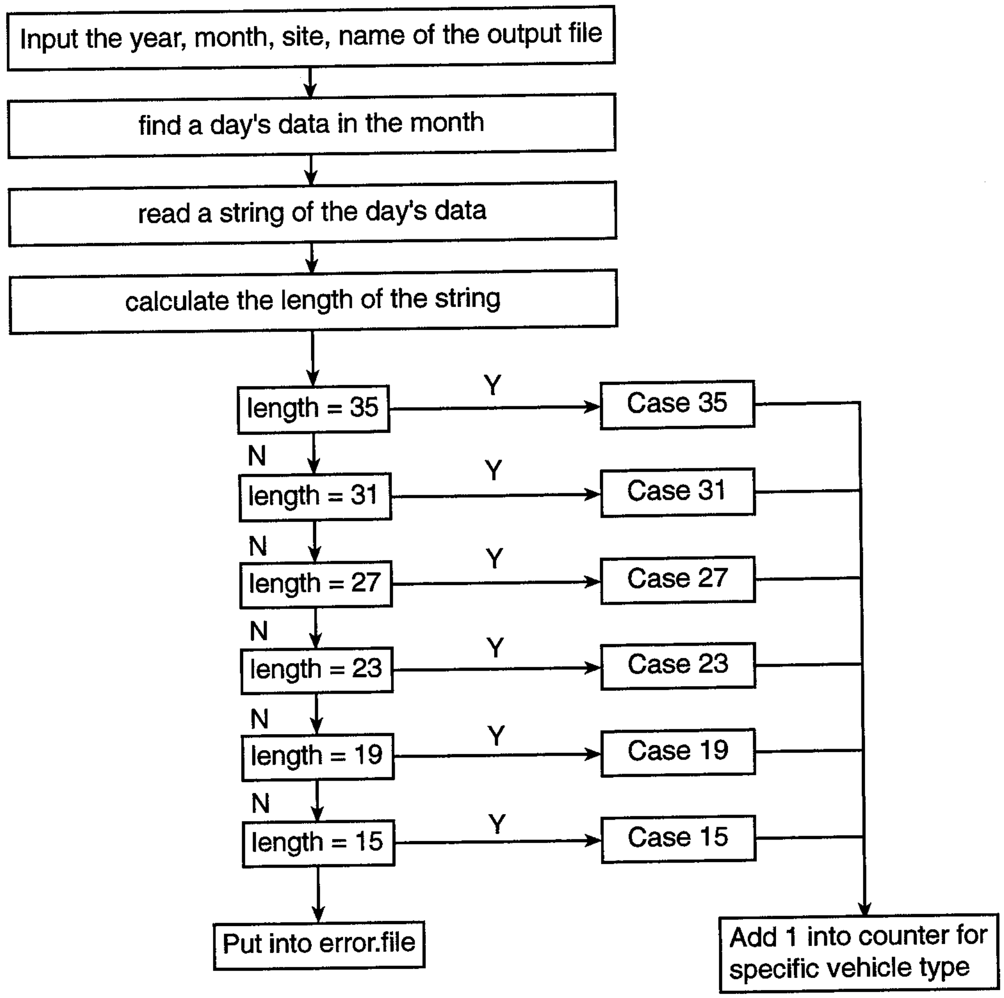


Figure 2.4 Vehicle sorting program flow chart

Combined error — Sometimes the above-mentioned errors occurred together in one error record. For example, one common combined error was the occurrence of two vehicles in one record, with one or both of them being off scale. Typical examples of error records are:

- Off scale
  A 5-axle single trailer (line 7428 of v0010809.95) with the right wheel of its last axle off scale:
  2,8,8,23,1,3,69,42,18,3.3,4.4,29,3.0,0.8,<u>18.0</u>,22,3.5,0.9,<u>4.6</u>,4,3.0,1.1,<u>34.6</u>,
  25,2.6,0.0,<u>41.8</u>

A 2-axle passenger car (line 3156 of v0020909.95) with the right wheel of its second axle off scale:
1,9,8,12,32,46,56,0,0,1.0,0.7,0,0.8,0.0,72.8

- Several vehicles in one record
  A passenger car with axle spacing 4.5 ft followed by a 5-axle single trailer with axle spacing 10.7 - 4.5 - 37.4 - 4.3 ft (line 242 of v0021217.93):
  2,12,16,3,18,27,68,17,21,2.9,3.0,19,1.5,2.1,4.5,15,4.1,2.2,507.6,15,2.7,4.0,10.7,15,2.4,4.1,4.5,18,3.3,2.7,37.4,18,3.5,2.0,4.3

- Unreasonable axle spacing
  The second axle spacing (56.6 ft) appears unreasonable for any 6-axle truck (line 1214 of v0010809.95):
  1,8,8,7,40,57,69,20,18,2.9,4.2,23,6.2,6.5,12.3,15,2.0,1.5,56.6,13,2.4,1.6,3.3,14,2.4,1.7,2.9,13,2.7,1.8,2.9

- Error in axle spacing compared with the TxDOT classification
  Both the first and the third axle spacing of a 3S1 single trailer with axle spacing 23.0 - 4.8 - 40.5 ft are slightly over the standard range with the first axle spacing 6.1 - 20.0 ft, the second one 3.4 - 6.0 ft, and the third one 6.1 - 40.0 ft (line 4783 of v0010324.94)
  1,3,23,16,11,54,83,22,11,4.5,5.0,14,3.4,4.4,23.0,13,3.5,4.6,4.8,13,2.4,2.4, 40.5

- Combined error
  A 5-axle single trailer with axle spacing 13.7 - 4.2 - 30.9 - 4.1 ft connected with a 4-axle truck with the second truck off scale (line 1381 of v0010809.95)
  1,8,8,8,16,16,59,19,22,4.6,6.5,22,3.0,3.5,13.7,24,2.5,3.4,4.2,38,2.1,2.4,30.9,31,1.1,2.6,4.1,21,6.4,0.9,98.0,34,3.6,0.7,9.9,26,3.2,0.0,492.9,21,1.6,0.0, 32.8

Error files for four days were chosen randomly from different years and sites to analyze the composition of these errors. Table 2.2 details these four files.

2.4 ANALYSIS OF OBSERVED TRAFFIC

Data from 1995 were chosen to represent the present traffic situation at the WIM sites. Owing to the inconsistent data recorded in July and August of 1995, the error records of these two months were about 10 times higher than those of any other time. Therefore, the data from July and August are not used in the analysis.

2.4.1 Traffic Composition

Table 2.3 summarizes all thirteen types of vehicles and error files recorded for each month of 1995. Of all vehicles, motorcycles, 2-axle passenger cars, pickup trucks and busses

(Type 1, Type 2 and Type 3) accounted for 73 percent, and trucks accounted for the remaining 27 percent. Of the trucks, 59 percent were 5-axle single trailers (Type 9), 26 percent were 2-axle 6-tire single unit (with or without 1 or 2 axle trailer), 5 percent were 4-axle semi-trailer (Type 8), and the rest of the trucks accounted for 10 percent. Figure 2.5 and Figure 2.6 show traffic composition and truck composition, respectively.

Table 2.2 Error file analysis

Vehicle/day	V0021217.93	V0010324.94	V0010809.95	V0020909.95
Total vehicles	7481	6403	7593	8047
Error records	83	11	53	39
Percentage of total vehicles	1%	0.1%	0.7%	0.5%
Items of error records:				
Off scale (Percentage of error records)	21 (25%)	-	17 (32%)	8 (20%)
Several vehicles in one record	5 (7%)	1 (10%)	16 (30%)	12 (31%)
Unreasonable axle spacing	7 (10%)	7 (63%)	17 (32%)	12 (31%)
Error in axle spacing	4 (7%)	3 (27%)	3 (6%)	7 (18%)
Ghost record	46 (50%)	-	-	-

Table 2.3 Summary of south site vehicle count by class (1995)

Month	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	Type 7	Type 8	Type 9	Type 10	Type 11	Type 12	Type 13	Type 14	Total
Jan.	86084	56248	788	14621	1544	20	907	2641	33996	423	1584	260	19	723	199858
Feb.	80613	52739	869	14943	1768	12	926	2611	32710	415	1606	267	12	765	190256
Mar.	99229	61765	1024	15672	1722	16	1249	2946	37006	558	1866	309	13	863	224238
Apr.	72838	44937	766	12378	1195	18	848	2013	24365	337	1268	181	9	658	161811
May	64806	38587	603	9798	1384	15	718	1864	21119	319	999	129	13	539	140893
Jun.	26874	16438	219	4499	770	4	324	847	11451	157	416	62	7	267	62335
Sep.	101340	53558	897	15497	2747	13	995	3073	34568	574	1673	297	18	971	216221
Oct.	110043	45855	791	13812	2388	13	934	3025	33937	497	1646	305	21	725	213992
Nov.	106337	42297	707	12473	1806	6	686	2441	27583	345	1227	265	8	609	196790
Dec.	109581	41226	641	12041	1635	7	594	2468	26261	413	1283	242	13	531	196936
Sum	857745	453650	7305	125734	16959	124	8181	23929	282996	4038	13568	2317	133	6651	2E+06
%	**48%**	**25%**	**0%**	**7%**	**1%**	**0%**	**0%**	**1%**	**16%**	**0%**	**1%**	**0%**	**0%**	**0%**	**100%**
Truck Percentage				125734 **26.3%**	16959 **3.5%**	124 **0.0%**	8181 **1.7%**	23929 **5.0%**	282996 **59.2%**	4038 **0.8%**	13568 **2.8%**	2317 **0.5%**	133 **0.0%**		477979

Note: Data deficient in some months

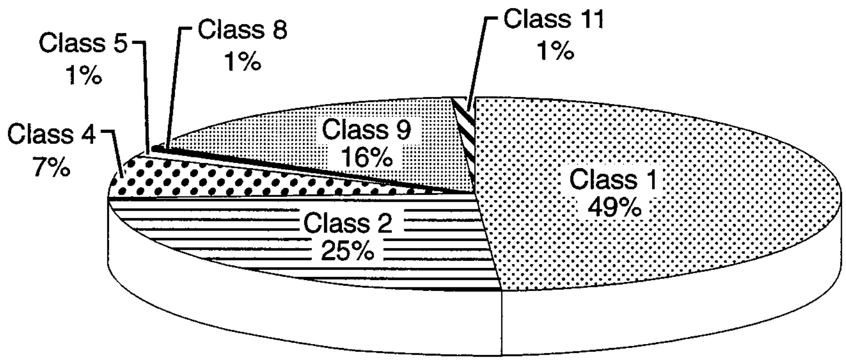


Figure 2.5 Traffic composition (1995)

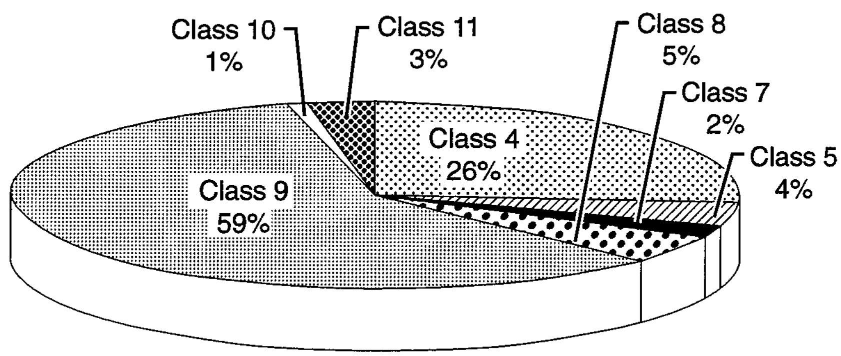


Figure 2.6 Truck composition (Classes 6, 12, and 13=0%) (1995)

2.4.2 North and South Site Traffic Comparison

As stated in Section 2.1, two WIM systems are installed north and south of Corrigan on US 59. US 287 intersects US 59 at Corrigan. Since both WIM systems are installed on the southbound lanes of US 59, vehicles coming from the north and traveling through Corrigan, along with southbound vehicles turning to and from US 287, can be observed. Figure 2.7 compares traffic at the two sites recorded from September 1 to October 30, 1995. There is a consistent pattern for north and south site traffic, as shown in the figure. During weekdays, the traffic count at the north site is always slightly higher than that at the south site. The traffic count is almost identical on the weekend (Saturday and Sunday). Thus, most of the traffic traveled through Corrigan, with a very small percentage of traffic using US 287 on weekdays.

2.4.3 Traffic Lanewise Distribution

Traffic lanewise distribution is also an important factor in pavement design. Unequal distribution of traffic between lanes significantly affects pavement design and performance. Table 2.4 summarizes the lanewise distribution of vehicle classes. As shown in Figure 2.8, Lane 1 and Lane 2 are the right-hand lane and the left-hand lane, respectively. From Table 2.4 and Figure 2.8, it can be observed that passenger cars (Type 1), pickups (Type 2), and small trucks (Type 4) used the left-hand lane more often than heavy trucks and busses (Type 3). In general, 72 percent of the vehicles traveled in the right-hand lane and 28 percent in the left-hand lane. Moreover, 80 percent of the trucks traveled in the right-hand lane.

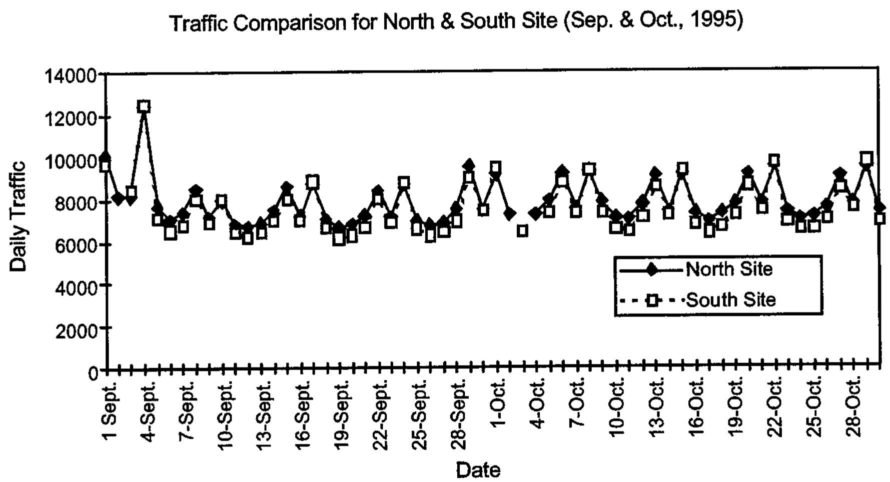


Figure 2.7 Traffic comparison for north and south WIM site

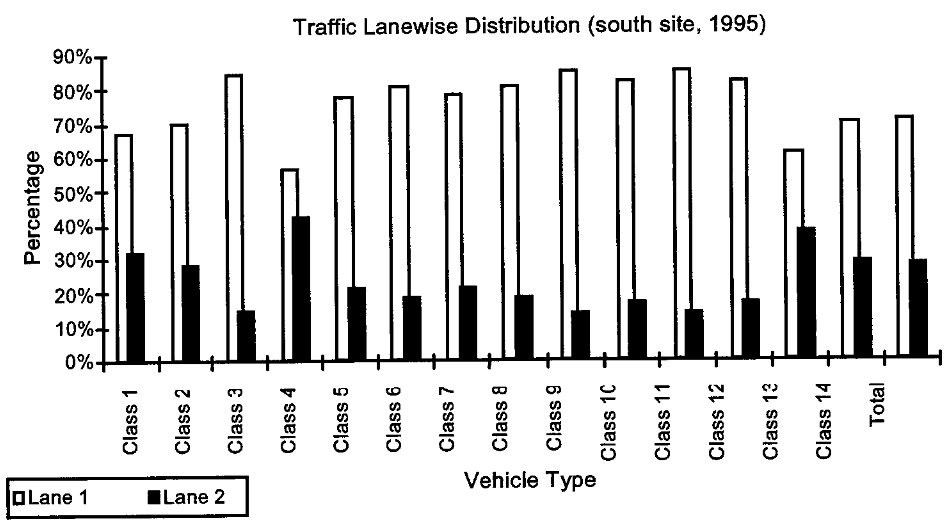


Figure 2.8 Lanewise traffic distribution

Table 2.4 Traffic lanewise distribution of south site (Jan., Feb., Mar. of 1995)

Month	**Site (Lane)**	**Type 1**	**Type 2**	**Type 3**	**Type 4**	**Type 5**	**Type 6**	**Type 7**	**Type 8**	**Type 9**	**Type 10**	**Type 11**	**Type 12**	**Type 13**	**Type 14**	**Total**
Jan.	1	59326	40314	675	8573	1223	16	733	2239	29541	356	1375	228	14	542	1E+05
	2	26758	15934	113	6048	321	4	174	402	4455	67	209	32	5	181	54703
Feb.	1	55513	37786	733	8297	1413	9	717	2123	28307	334	1367	210	7	521	1E+05
	2	25100	14953	136	6646	355	3	209	488	4403	81	239	57	5	244	52919
Mar.	1	65449	43139	864	9094	1291	14	969	2298	31220	462	1596	254	6	604	2E+05
	2	33780	18626	160	6578	431	2	280	648	5786	96	270	55	7	259	66978
sum	1	2E+05	1E+05	2272	25964	3927	39	2419	6660	89068	1152	4338	692	27	1667	4E+05
	2	85638	49513	409	19272	1107	9	663	1538	14644	244	718	144	17	684	2E+05
Total		3E+05	2E+05	2681	45236	5034	48	3082	8198	1E+05	1396	5056	836	44	2351	6E+05
%	1	68%	71%	85%	57%	78%	81%	78%	81%	86%	83%	86%	83%	61%	71%	72%
	2	32%	29%	15%	43%	22%	19%	22%	19%	14%	17%	14%	17%	39%	29%	28%

CHAPTER 3. TRAFFIC FORECASTING

Traffic data are one of the key requirements for pavement design. Using such data, pavement engineers forecast future traffic loads and then design a given pavement structure to withstand these loads (and to prevent premature failures). In this study, we analyzed a separate growth rate for forecasting the number of trucks in each of several classes. Emphasis is placed on 5-axle single trailer trucks. A linear regression traffic forecasting method and a time-series method are used to analyze historical traffic count data obtained from TxDOT survey records and data from two research WIM systems, respectively.

3.1 LINEAR REGRESSION ON HISTORICAL SURVEY DATA FROM TXDOT

TxDOT has two manual-observation traffic stations: Station 1285 north and 1285 south, near Corrigan. At these stations, 24-hour non-direction classification manual counts are adjusted to represent average annual daily counts. These data are available from 1986 to 1994, with some data missing for 1987 and 1989. Scatter diagrams were plotted for the 24-hour non-direction classification counts of all truck classes; it was noted that only 5-axle single trailers showed a strong increasing linear trend, as seen in Figure 3.1.

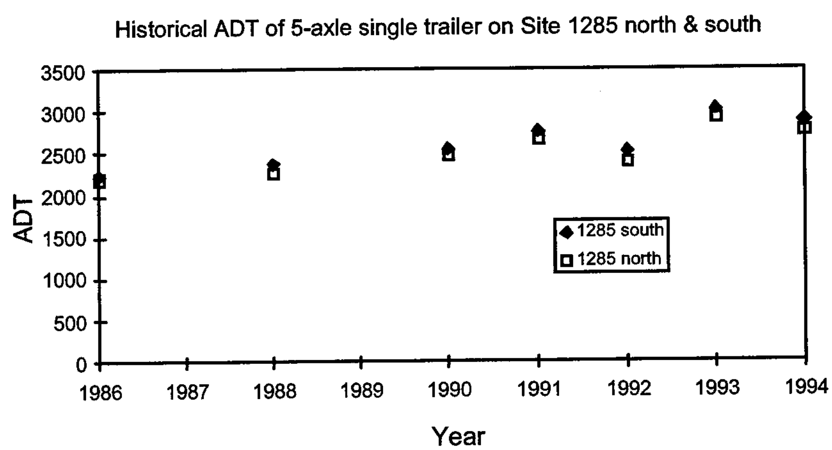


Figure 3.1 Twenty-four-hour non-direction count of 5-axle single trailer (1986–1994)

The following linear regression form was used in this study to evaluate traffic growth rates:

$$Y = a + bX$$

where

a = the Y intercept (constant),
b = the slope, or the rate of change of Y,
Y = dependent variable (24-hour count by class), and
X = independent variable (year).

The results of regression analysis are:

Station 1285 north:

$$Y = -158468 + 81 * X$$

Station 1285 south:

$$Y = -173902 + 89 * X$$

Both stations (1285 north and south) showed good fit with the data. R^2, the coefficient of determination, is a summary measure that tells how well the sample regression line fits the data (Ref 10). The value of R^2 lies between 0 and 1, with $R^2 = 1$ representing a perfect fit. In this study, the R^2 value was above 0.8 for both stations, which means that more than 80 percent of the total 24-hour non-direction count variation of 5-axle single trailers can be explained by the regression model.

According to the regressed linear equation, the average annual traffic growth rate for 5-axle single trailers can be obtained by $(Y_{1994}-Y_{1986})/(8*Y_{1986})$. The results were about 4 percent for both station 1285 north and south.

3.2 TIME SERIES METHOD FOR FORECASTING 5-AXLE SINGLE TRAILER TRAFFIC

A time series is "a chronological sequence of observations on a particular variable" (Ref 5). In this study, average daily truck count by class is a time series. The time-series method is used in an attempt to discover a historical pattern that can be exploited in the preparation of a forecast. A time series comprises the following components:

1. Trend
2. Cycle
3. Seasonal variations
4. Irregular fluctuations

Trend refers to the upward or downward movement that a time series has over a period of time. Thus, trend reflects the long-term growth or decline in the time series.

Cycle refers to recurring up and down movements around trend levels. These fluctuations measured from peak to peak can have a duration of anywhere from 2 to 10 years, or even longer.

Seasonal variations are periodic patterns in a time series that complete themselves within a calendar year and are repeated on a yearly basis. Seasonal variations are usually caused by weather or other local factors.

Irregular fluctuations are erratic movements in a time series that follow no recognizable or regular pattern.

3.2.1 Exploring Historical Pattern of WIM Data

Daily counts of Type 9 (5-axle single trailer) and Type 4 (2-axle/ 6 tire single unit) trucks from January 1 to April 22, 1995, were plotted to show a pattern (see Figure 3.2). The 5-axle single trailer (Type 9) trucks showed a consistent weekly pattern. The lowest Type 9 counts were observed on Saturdays and Sundays. On weekdays, these counts were always lowest on Fridays and highest on Thursdays.

It was, however, not considered necessary to use daily traffic as a time series to forecast traffic for a 20-year period. To do so would necessitate calculating 365 seasonal factors for each day in a year, providing short-term fluctuations that are of no interest. Nevertheless, the consistent weekly pattern was deemed useful. Equipment malfunctions resulted in short periods of time during which data were not recorded. The result shown in Section 2.4.2 was used to supply the missing data, since the traffic counts between the two WIM sites were almost identical. If the data from both sites were missing, the consistent weekly pattern was used to estimate the missing data values by using the data set of the closest adjacent week. After the data gaps were filled, the monthly data of three years (1993–1995) were plotted to explore the monthly pattern, as shown in Figure 3.3. By inspection, a linear, increasing trend can be observed in the data shown in Figure 3.3, and an increasing seasonal variation can be seen in Figure 3.4.

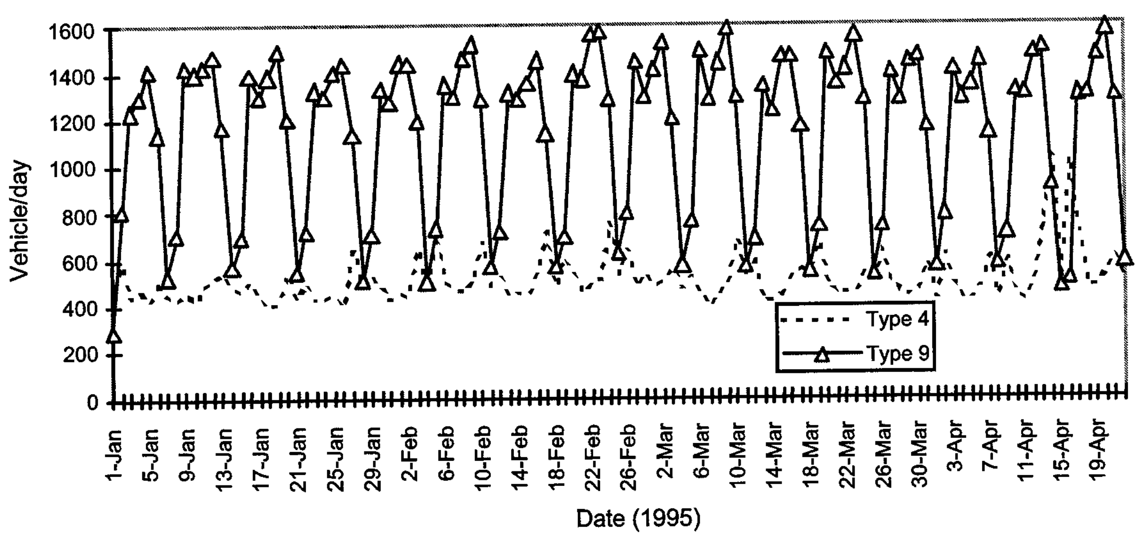


Figure 3.2 Daily vehicle count

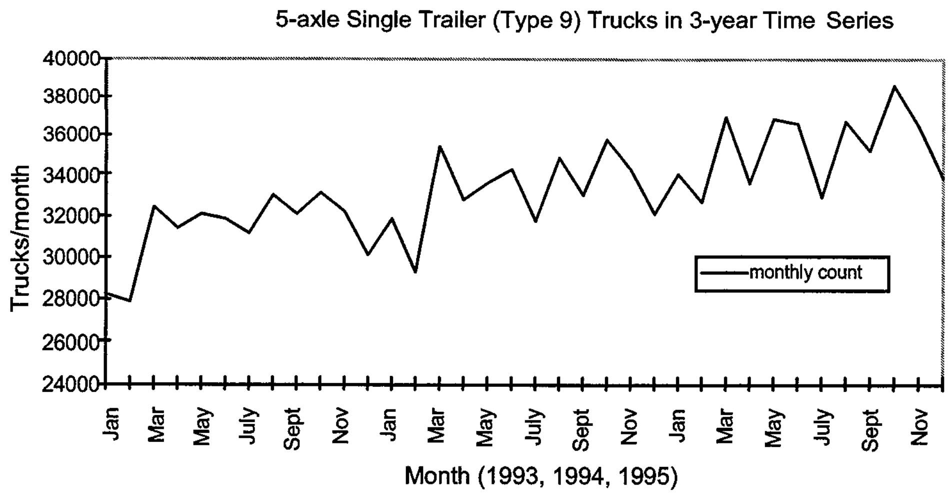


Figure 3.3 Monthly count of 5-axle single trailer trucks, 1993–1995

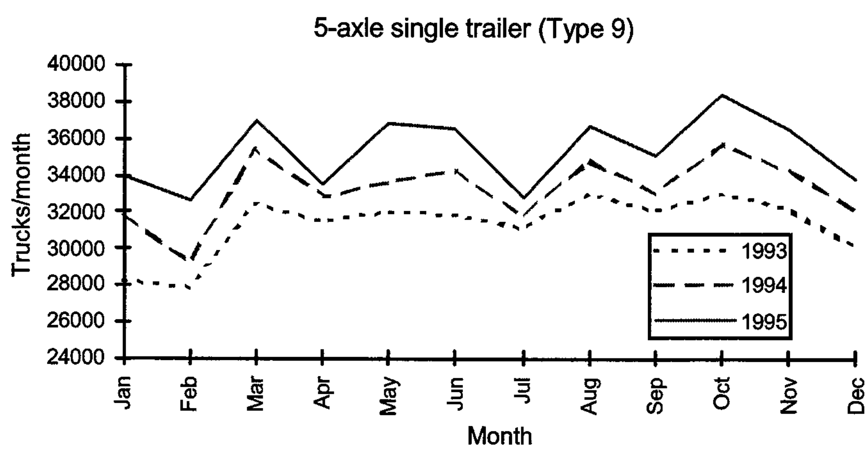


Figure 3.4 Monthly count of 5-axle single trailer trucks over 3 years (1993–1995)

3.2.2 Time Series Decomposition

Among the time-series models, one model that isolates the trend effects and seasonal effects is the multiplicative decomposition model. This model is an intuitive approach of the time series that displays increasing or decreasing seasonal variation. In this study, it can be observed in Figure 3.3 and Figure 3.4 that the data show an increasing annual truck count during the three-year period that was studied. The multiplicative decomposition model can be expressed as follows:

$$Y_t = TR_t * SN_t * CL_t * IR_t$$

where

Y_t = the observed value of the time series in time period t,

TR_t = the trend component (or factor) in time period t,

SN_t = the seasonal component (or factor) in time period t,

CL_t = the cyclical component (or factor) in time period t, and

IR_t = the irregular component (or factor) in time period t.

The first step of this time-series analysis is to calculate the seasonal factor (SN_t). Moving averages and centered moving averages (CMA) are used to attenuate short-term variations and irregular data fluctuations. The first moving average value is the monthly average of the first 12 months' count of 5-axle single trailer trucks. The second moving average value is obtained by dropping the first-month count (Y_1) from the average and including the thirteenth-month count (Y_{13}). Successive moving averages are computed similarly until the last-month count Y_{36} is included in the last moving average (see Table 3.1 for the worksheet). The first moving average value corresponds to a time that is midway between periods 6 and 7; the second value corresponds to a time that is midway between periods 7 and 8; and so forth. In order to obtain averages corresponding to time periods in the original monthly truck count time series, the CMA needs to be calculated. The CMA is a two-period moving average of the previously-computed twelve-period moving averages. Thus the first CMA is the average of the first and the second moving average values. The second CMA is the average of the second and the third values. Successive CMAs are calculated in a similar fashion.

The CMA in time period t, CMA_t, is considered equal to the estimate of $TR_t * CL_t$, because the averaging procedure is assumed to have removed seasonal variations (SN_t) and irregular fluctuations (IR_t), though the trend effects and cyclical effects still remain. Since the model

$$Y_t = TR_t * SN_t * CL_t * IR_t$$

implies that

$$SN_t * IR_t = \frac{Y_t}{TR_t * CL_t} = \frac{Y_t}{CMA_t}$$

The estimate SN_t can be found by the normalization process:

$$SN_t = N_f * \overline{SN_t}$$

$\overline{SN_t}$ can be obtained by grouping the values of SN_t * IR_t by months and calculating an average for each month. N_f is the normalization factor and was calculated by:

$$L \Big/ \sum_{t=1}^{L} \overline{SN_t} = 12 / 12.019 = 0.9984$$

Table 3.1 Analysis of the monthly count of 5-axle single trailer trucks by the time-series method

t	Y_t	12 period moving average	$CMA_t = TR_t * CL_t$	$SN_t * IR_t = Y_t/(TR_t * CL_t)$	SN_t	$D_t = Y_t/SN_t$	$TR_t = 30161.05+171.742t$	$TR_t * SN_t$
1	28270				0.983	28769.667	30332.792	29805.977
2	27861				0.922	30224.93	30504.534	28118.736
3	32450				1.056	30732.077	30676.276	32391.08
4	31437				0.979	32095.153	30848.018	30215.439
5	32051				1.032	31042.719	31019.76	32027.296
6	31812	31287.833			1.034	30778.64	31191.502	32238.723
7	31147	31582.000	31434.917	0.991	0.971	32093.01	31363.244	30438.745
8	33026	31698.500	31640.250	1.044	1.039	31774.128	31534.986	32777.436
9	32078	31948.250	31823.375	1.008	0.991	32372.434	31706.728	31418.349
10	33080	32062.833	32005.542	1.034	1.043	31716.546	31878.47	33248.884
11	32165	32196.333	32129.583	1.001	1.003	32078.792	32050.212	32136.344
12	30077	32400.583	32298.458	0.931	0.932	32283.88	32221.954	30019.308
13	31800	32453.833	32427.208	0.981	0.983	32362.059	32393.696	31831.088
14	29259	32607.083	32530.458	0.899	0.922	31741.547	32565.438	30018.454
15	35447	32684.667	32645.875	1.081	1.056	33570.414	32737.18	34567.188
16	32812	32909.000	32796.833	1.000	0.979	33498.94	32908.922	32234.081
17	33653	33087.000	32998.000	1.020	1.032	32594.322	33080.664	34155.139
18	34263	33251.333	33169.167	1.033	1.034	33150.024	33252.406	34368.82
19	31786	33434.333	33342.833	0.953	0.971	32751.418	33424.148	32438.9
20	34865	33721.917	33578.125	1.038	1.039	33543.42	33595.89	34919.538
21	33009	33851.833	33786.875	0.977	0.991	33311.979	33767.632	33460.508
22	35772	33914.750	33883.292	1.056	1.043	34297.591	33939.374	35398.384
23	34301	34178.000	34046.375	1.007	1.003	34209.067	34111.116	34202.786
24	32049	34373.000	34275.500	0.935	0.932	34400.574	34282.858	31939.331
25	33996	34462.333	34417.667	0.988	0.983	34596.873	34454.6	33856.198
26	32710	34612.333	34537.333	0.947	0.922	35485.355	34626.342	31918.172
27	37006	34791.417	34701.875	1.066	1.056	35046.879	34798.084	36743.297
28	33567	35019.333	34905.375	0.962	0.979	34269.746	34969.826	34252.724
29	36812	35204.917	35112.125	1.048	1.032	35653.944	35141.568	36282.982
30	36603	35356.750	35280.833	1.037	1.034	35414.013	35313.31	36498.916
31	32858				0.971	33855.977	35485.052	34439.054
32	36665				1.039	35275.19	35656.794	37061.639
33	35158				0.991	35480.704	35828.536	35502.668
34	38507				1.043	36919.863	36000.278	37547.884
35	36528				1.003	36430.098	36172.02	36269.228
36	33871				0.932	36356.262	36343.762	33859.355

These calculations are summarized in Table 3.2.

Table 3.2 Estimates of the seasonal factor for 5-axle trailer trucks

		$SN_t*IR_t=Y_t/(TR_t*CL_t)$			
		Year 1	Year 2	$\overline{SN_t}$	$SN_t=0.998(\overline{SN_t})$
1	Jan.	0.981	0.988	0.984	0.983
2	Feb	0.899	0.947	0.923	0.922
3	Mar	1.081	1.066	1.074	1.072
4	Apr	1.000	0.962	0.981	0.979
5	May	1.020	1.048	1.034	1.032
6	Jun	1.033	1.037	1.035	1.034
7	Jul	0.991	0.953	0.972	0.971
8	Aug	1.044	1.038	1.041	1.039
9	Sep	1.008	0.977	0.992	0.991
10	Oct	1.034	1.056	1.045	1.043
11	Nov	1.001	1.007	1.004	1.003
12	Dec	0.931	0.935	0.933	0.932

The deseasonalized observation in time period t is defined as:

$$d_t = \frac{Y_t}{SN_t}$$

The deseasonalized observations were calculated and are plotted in Figure 3.5.

Since the deseasonalized observations plotted in Figure 3.5 have a general straight-line appearance, it seems reasonable to assume that

$$TR_t = b_0 + b_1 t$$

TR_t was obtained by computing

$$b_1 = \frac{36\sum_{t=1}^{36} td_t - \left(\sum_{t=1}^{36} t\right)\left(\sum_{t=1}^{36} d_t\right)}{36\sum_{t=1}^{36} t^2 - \left(\sum_{t=1}^{36} t\right)^2} = 172$$

and

$$b_0 = \frac{\sum_{t=1}^{36} d_t}{36} - b_1\left(\frac{\sum_{t=1}^{36} t}{36}\right) = 30161$$

which gave

$$TR_t \;=\; b_0 + b_1 t = 30161 + 172\,t$$

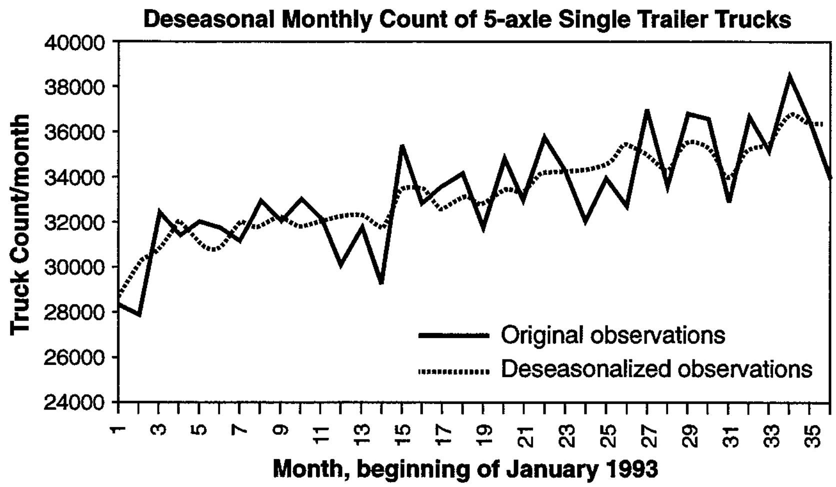


Figure 3.5 Deseasonalized monthly count of 5-axle single trailer trucks

After SN_t and TR_t were calculated, the next step was to estimate CL_t. Since the model implies that

$$CL_t * IR_t = \frac{Y_t}{TR_t * SN_t}$$

and CL_t can be estimated by averaging the irregular influence.

$$CL_t \;=\; \frac{CL_{t-1}IR_{t-1} + CL_t IR_t + CL_{t+1}IR_{t+1}}{3}$$

That is, CL_t is a three-period moving average of the CL_t*IR_t values. Finally, IR_t can be calculated by the equation

$$IR_t = \frac{CL_t * IR_t}{CL_t}$$

The calculated values of CL_t and IR_t are summarized in Table 3.3. Since data for only three years were available, and since most of the values of CL_t were near 1, it can be concluded that no well-defined cycle existed in the monthly data. Furthermore, the values of IR_t were examined, and there was also no pattern detected in the irregular factors.

3.2.3 Traffic Forecasting by a Time-Series Model

The model obtained using the multiplicative decomposition method can be used not only for describing the time series, but also for forecasting future time-series values.

Since there is no pattern in the irregular component in the model, it is reasonable to assume this component is equal to 1. Moreover, a well-defined cycle does not exist or cannot be predicted; therefore, the model for this data set can be written as

$$\hat{Y}_t = TR_t * SN_t = (30161 + 172\,t) * SN_t \text{ (Truck count/month)}$$

The forecasted count of 5-axle single trailer trucks in the first five months of 1996 was calculated by this forecasting model (see Table 3.4). The 100(1-α)% confidence interval for $\hat{Y}_t$ is

$$[\hat{Y}_t \pm t_{\alpha/2}^{(n-1)} s\sqrt{1+(1/n)}]$$

where

$$s = \sqrt{\frac{\sum_{t=1}^{n}\left(Y_t - \hat{Y}_t\right)^2}{n-1}}$$

and $t_{(\alpha/2)}^{(n-1)}$ is the point on the scale of the t-distribution having (n-1) degrees of freedom so that the area under the curve of this t-distribution to the right of $t_{(\alpha/2)}^{(n-1)}$ is α/2. If a 95 percent confidence interval, α = 0.05, is needed, then $t_{(\alpha/2)}^{(n-1)} = t_{(0.05/2)}^{(36-1)} = t_{(0.025)}^{35} \approx 2$. The value of s (standard deviation) for the 1993, 1994, 1995 data (n=36) is equal to 612. Therefore, a 95 percent confidence interval for the 5-axle single trailer truck count ($\hat{Y}$) at this site in a future time period is

$$[\hat{Y}_t \pm t_{\alpha/2}^{(n-1)} s\sqrt{1+(1/n)}] = [\hat{Y}_t \pm 2(612)\sqrt{1+(1/36)}] = [\hat{Y}_t \pm 1241]$$

Table 3.3 Estimates of the cycle and irregular factor for 5-axle single trailer trucks

t	Y_t	TR_t * SN_t	CL_t*IR_t=Y_t/(TR_t*SN_t)	CL_t	IR_t=(CL_t*IR_t)/CL_t
1	28270	29805.977	0.9485		
2	27861	28118.736	0.9908	0.9804	1.0107
3	32450	32391.08	1.0018	1.0110	0.9909
4	31437	30215.439	1.0404	1.0143	1.0257
5	32051	32027.296	1.0007	1.0093	0.9915
6	31812	32238.723	0.9868	1.0036	0.9832
7	31147	30438.745	1.0233	1.0059	1.0173
8	33026	32777.436	1.0076	1.0173	0.9905
9	32078	31418.349	1.0210	1.0078	1.0131
10	33080	33248.884	0.9949	1.0056	0.9894
11	32165	32136.344	1.0009	0.9992	1.0016
12	30077	30019.308	1.0019	1.0006	1.0013
13	31800	31831.088	0.9990	0.9919	1.0072
14	29259	30018.454	0.9747	0.9997	0.9750
15	35447	34567.188	1.0255	1.0060	1.0193
16	32812	32234.081	1.0179	1.0096	1.0083
17	33653	34155.139	0.9853	1.0000	0.9852
18	34263	34368.82	0.9969	0.9874	1.0097
19	31786	32438.9	0.9799	0.9917	0.9880
20	34865	34919.538	0.9984	0.9883	1.0103
21	33009	33460.508	0.9865	0.9985	0.9880
22	35772	35398.384	1.0106	1.0000	1.0106
23	34301	34202.786	1.0029	1.0056	0.9973
24	32049	31939.331	1.0034	1.0035	1.0000
25	33996	33856.198	1.0041	1.0108	0.9934
26	32710	31918.172	1.0248	1.0120	1.0126
27	37006	36743.297	1.0071	1.0040	1.0032
28	33567	34252.724	0.9800	1.0006	0.9794
29	36812	36282.982	1.0146	0.9991	1.0155
30	36603	36498.916	1.0029	0.9905	1.0125
31	32858	34439.054	0.9541	0.9821	0.9715
32	36665	37061.639	0.9893	0.9779	1.0117
33	35158	35502.668	0.9903	1.0017	0.9886
34	38507	37547.884	1.0255	1.0077	1.0178
35	36528	36269.228	1.0071	1.0110	0.9962
36	33871	33859.355	1.0003		

The actual traffic data for the first five months of 1996 were used to test this model. The 95 percent confidence intervals (calculated by the above method) for the 5-axle single trailer truck count for the first five months of 1996 are presented in Table 3.4. Note that four of the five-month confidence intervals contain the actual traffic values of the first five months of 1996 and that the fifth monthly count is only very slightly outside. The observed and estimated counts for 1993, 1994, 1995, and the first five months of 1996 are plotted in Figure 3.6.

According to the trend component in this time series model,

$$TR_t \;=\; b_0 + b_1 t = 30161 + 172\ t \text{ (Truck count/month)},$$

and the amount of annual increase for 5-axle single trailer trucks is 172 * 12. Therefore, during the 41-month period beginning in January 1993, the annual growth rate for 5-axle single trailer trucks is the amount of annual increase divided by the first month count, which is about 7 percent.

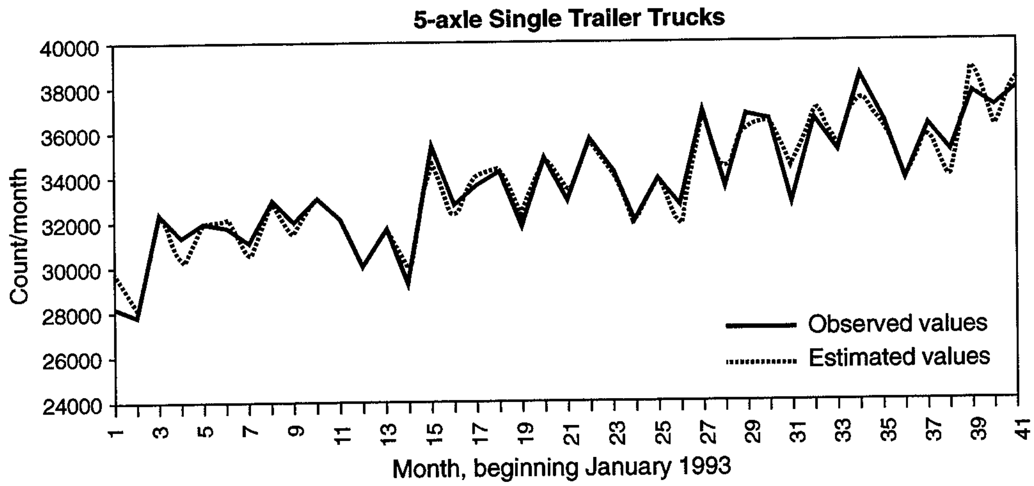


Figure 3.6 Estimated and observed values of 5-axle single trailer truck

Table 3.4 Forecasted and observed 1996 counts of 5-axle single trailer trucks

t	SN_t	TR_t= 30161.05+171.742t	TR_t*SN_t	$B_t(95)$	[$TR_t*SN_t-B_t(95)$, $TR_t*SN_t+B_t(95)$]	1996 Count
37	0.983	36515.504	35881.309	1241.309	[34640, 37122]	36374
38	0.922	36687.246	33817.890	1241.309	[32576, 35059]	35079
39	1.056	36858.988	38919.405	1241.309	[37678, 40160]	37751
40	0.979	37030.73	36271.366	1241.309	[35030, 37512]	37083
41	1.032	37202.472	38410.825	1241.309	[37169, 39652]	37919

Note: $B_t(95)$ is the error bound in a 95 percent confidence interval.

3.3 TRAFFIC GROWTH RATE FOR OTHER TRUCK CLASSES

In addition to 5-axle single trailers, which accounted for about 60 percent of the total truck traffic at the WIM site in 1995, 2-axle/6 tire single units (2D) contributed approximately 26 percent of the truck traffic. The daily counts of 2Ds in September and October for the same period of three years are plotted in Figure 3.7. The data for 2D trucks showed a weekly pattern that differed from the pattern for 5-axle single trailers (see Figure 3.2), but the percent increase in monthly count was almost the same (see Appendix C for 5-axle single trailer trucks and other truck types). Traffic growth rate and seasonal factor for 2D trucks can be obtained by making another time-series analysis similar to that described above for 5-axle single trailers. Although these calculations were not made, a visual analysis of the plotted data indicated that these two truck classes, 5-axle single trailer and 2D, have a similar growth rate at this site. Among other truck classes, Type 5 trucks (3-axle single unit), which account for 3.5 percent of the total truck traffic, displayed a clear weekly pattern but no increase in count; Type 8 trucks (4-axle single trailer), which account for 5 percent of the truck traffic, showed an unclear weekly pattern and a small decrease in count. All other truck classes — Type 6 trucks (4 or more axle single unit), Type 7 trucks (3-axle single trailer), Type 10 truck (6 or more single trailer), Type 11 trucks (5-axle multi-trailer), Type 12 trucks (6-axle multi-trailer), and Type 13 trucks (7 or more axle multi-trailer) — displayed no regular weekly pattern and no consistent change in count. The growth rates for, and proportions of, all truck classes are summarized in Table 3.5.

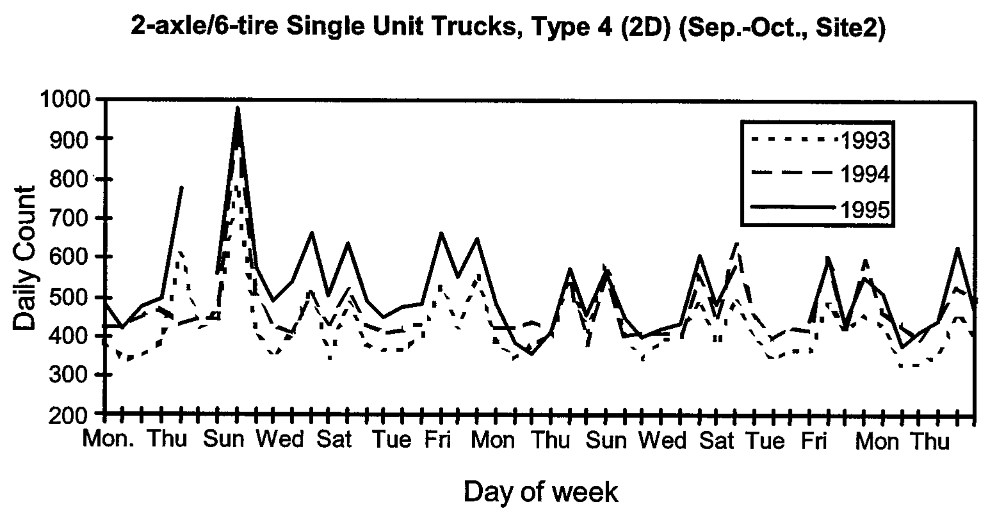


Figure 3.7 Daily count of 2D trucks in September and October, 1993–1995

3.4 SUMMARY

According to this study, the annual growth rate for 5-axle single trailer trucks from 1986 through 1994 was about 4 percent for the data set obtained from TxDOT's adjusted 24-hour non-direction counts using a linear regression method. The annual growth rate for southbound 5-axle single trailer trucks from 1993 through 1994 was about 7 percent using WIM data and a time-series method.

The data set from TxDOT was sampled from 24-hour non-directional classification manual counts. Data were available from 1986 to 1994, with some data missing for 1987 and 1989. WIM data from 1993 to 1995 were available for southbound traffic. The WIM data sample included almost every vehicle passing by the WIM site. The WIM data have a larger sample size but a shorter time span than the TxDOT data.

The advantage of the time-series method used in this study (compared with a linear regression method) is that both monthly change and the trend of change can be captured by the time series model; linear regression can only describe the trend.

Table 3.5 Composition and approximate change in count for all truck classes, 1993–1995

	Type 4	Type 5	Type 6	Type 7	Type 8	Type 9	Type 10	Type 11	Type 12	Type 13
Percentage of Trucks	26.30%	3.50%	0.00%	1.70%	5.00%	59.20%	0.80%	2.80%	0.50%	0.00%
Weekly Pattern	clear	clear	no	unclear	unclear	strong	unclear	clear	unclear	no
Change in Count	increase	no	no	no	decrease	increase	no	no	no	no
Annual Rate of Change	5%	0%	0%	0%	0%	6%	0%	0%	0%	0%

CHAPTER 4. OBSERVED TRUCK LOADING ANALYSIS

The analysis of observed truck loading is an important part of this study, inasmuch as it is used to forecast future truck loading. A computer program was developed in C language to process data from the PAT DAW100 WIM systems. Load frequency distributions for each axle group of each truck type were analyzed by using WIM data of 1993, 1994, and 1995. Comparisons were conducted among distributions between sites, years, and different axle groups.

4.1 COMPUTER PROGRAM FOR PROCESSING LOAD DATA

As stated in Section 2.3, vehicle wheel load information is recorded in an ASCII data file. An axle load is calculated by adding the wheel load on both ends of the axle. Based on an evaluation of the TxDOT classification scheme, it is desirable to amplify some of the axle groups. For example, the 3S2 and 2S3 truck classes are included in Type 9 by TxDOT classification. In addition, the 3S2 spread (Ref 18), which separates the last tandem of a 3S2 so as to act like two single axles, is also included in Type 9. The axle-group arrangements of these three truck classes are different. There is one steering axle, one drive-tandem axle, and one trailer-tandem axle on a 3S2; on a 2S3, there is one steering axle, one drive-single axle, and one tridem axle; and on a 3S2 spread, there is one steering axle, one drive-tandem axle, and two trailer-single axles.

In order to further classify trucks into each vehicle type, some condition judgments are used to amplify TxDOT's axle-spacing scheme (see Table 2.1 and Section 2.3.2). And 3S2 spread truck class was added into TxDOT classification scheme. The load interval for tandem axles and tridem axles is in 4-kip increments from 0 to 48 kip. The load interval for a single axle is in 2-kip increments from 0 to 24 kip. Pangburn's report (Ref 18) discussed the fact that the steering axles of some trucks are more than 12 kip. In the AASHO Road Test, the steering axle loads ranged from 2 to 12 kip and were not analyzed separately but were incorporated into the single-axle and tandem-axle load factors. However, the damage caused by steering-axle loads over 12 kip should not be ignored. The computer program described here only counts the steering-axle loads more than 12 kip and gives the frequency distribution with the loads ranging from 12 to 20 kip.

The axle group lists for all truck classes are listed below. The computer program computes the weight distributions of all the following axle groups. The letter "S" stands for the single axle group, "T" stands for the tandem axle group, "R" stands for the tridem axle group, and "E" stands for the steering axle. "DT" and "DS" were named, respectively, for the tandem and single axle group of 3S2 spread which was added into the TxDOT classification scheme. The first number in the footnote of the axle group stands for the vehicle type, while the second number indicates the number of repetitions of the same axle group. If there is only one repetition of the same axle group in any truck class, the second number is ignored.

- Type 4
  2D — S_{41}, E_{41}
  2D-1 axle trailer — S_{42}, S_{43}, E_{42}
  2D-2 axle trailer — S_{44}, T_4, E_{43}
- Type 5
  3-axle single unit (3A) — T_5, E_5
- Type 6
  4-axle single unit (4A) — R_6, E_{62}
  4-axle single unit (Rig) — S_6, T_6, E_{61}
- Type 7
  2S1 — S_{71}, S_{72}, E_7
- Type 8
  2S2 — S_{81}, T_{81}, E_{81}
  3S1 — T_{82}, S_{82}, E_{82}
- Type 9
  2S3 — S_9, R_9, E_{91}
  3S2 — T_{91}, T_{92}, E_{92}
  3S2 spread — DT_9, DS_{91}, DS_{92}, E_{93}
- Type 10
  3S3 — T_{10}, R_{10}, E_{10}
- Type 11
  2S1-2 — S_{111}, S_{112}, S_{113}, S_{114}, E_{11}
- Type 12
  2S2-2 — S_{1211}, T_{121}, S_{1212}, S_{1213}, E_{121}
  3S1-2 — T_{122}, S_{1221}, S_{1222}, S_{1223}, E_{122}
- Type 13
  3S2-2 — T_{131}, T_{132}, S_{131}, S_{132}, E_{13}

Appendix D contains an example of how to use the program to calculate axle-load frequency; an output file for March 23, 1995, is also provided.

4.2 COMPARISON OF LOAD DISTRIBUTION AMONG YEARS

Data for the years 1993, 1994, and 1995 were processed using the computer program. Load distributions of the two tandem axles of 3S2s were used as examples to explore the difference among years. Sample sizes (observations) are listed in Table 4.1.

The load distributions for all three years are shown in Figure 4.1 and Figure 4.2. It can be observed that the distributions of the three years are almost identical.

Table 4.1 Sample size of 3S2 trucks

Year	North Site	South Site
1993	222,196	212,681
1994	228,105	299,304
1995	247,643	278,395

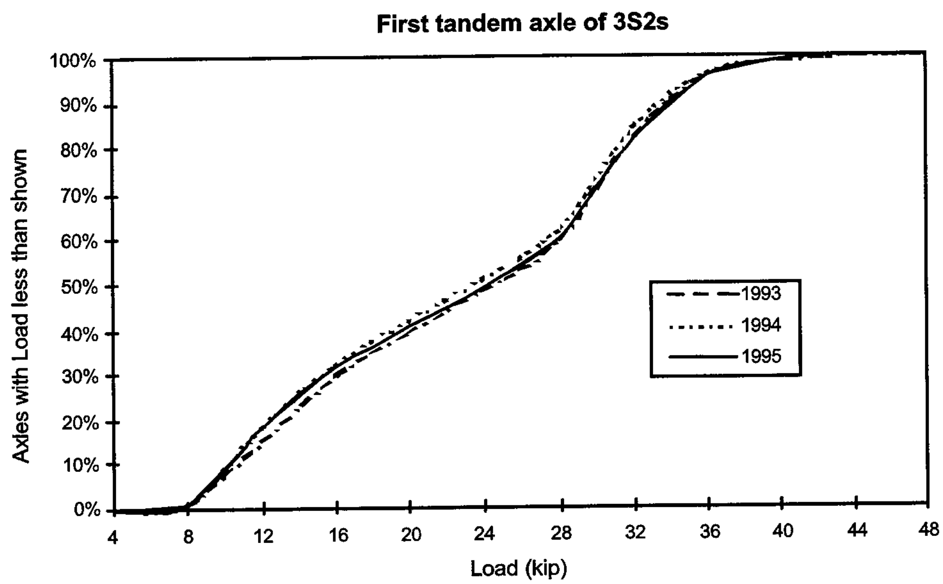


Figure 4.1 Load frequency distribution of the first tandem of 3S2s by year

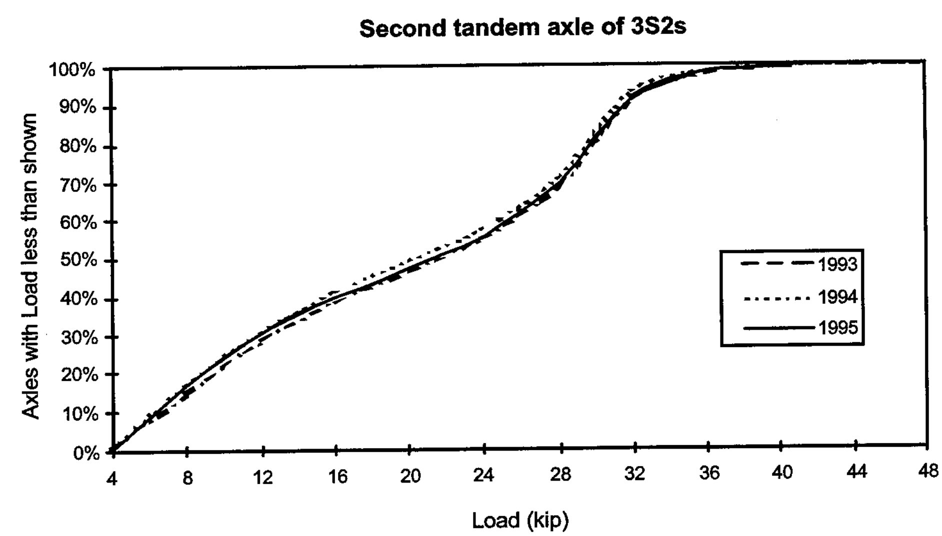


Figure 4.2 Load frequency distribution of the second tandem of 3S2s by year

4.3 COMPARISON OF LOAD DISTRIBUTION BETWEEN SITES

Comparison of the traffic between sites was discussed in Section 2.4.2. Most of the traffic went through both sites. However, some vehicles pass through only the north site and then turn at US 287. Figure 4.3 and Figure 4.4 show the differences in load distribution of the two tandem axles of 3S2 between two sites. Load distributions of two tandem axles of both sites were similar. Figure 4.5 shows the difference of the single axle of 2D between sites. The load distributions of the two sites were almost identical.

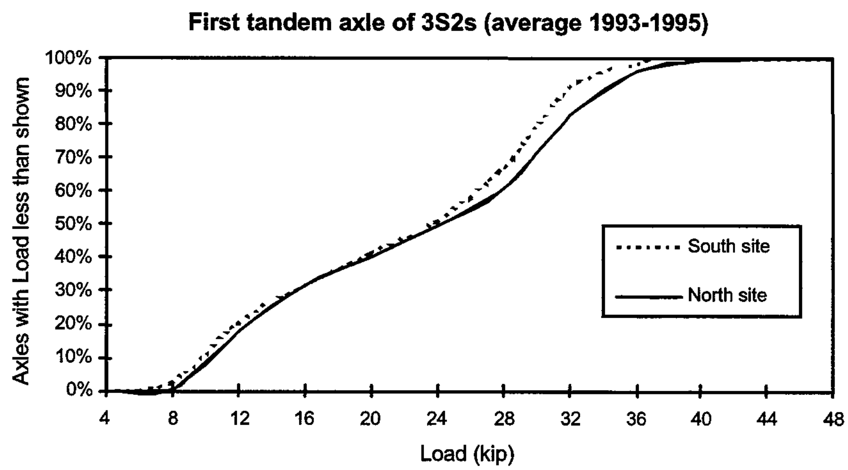


Figure 4.3 Load frequency distribution of the first tandem of 3S2s at north and south sites

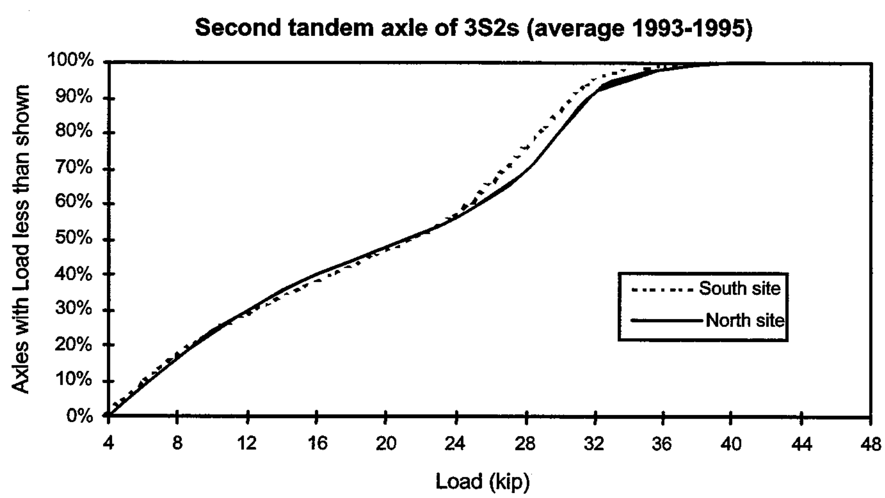


Figure 4.4 Load frequency distribution of the second tandem of 3S2s at north and south sites

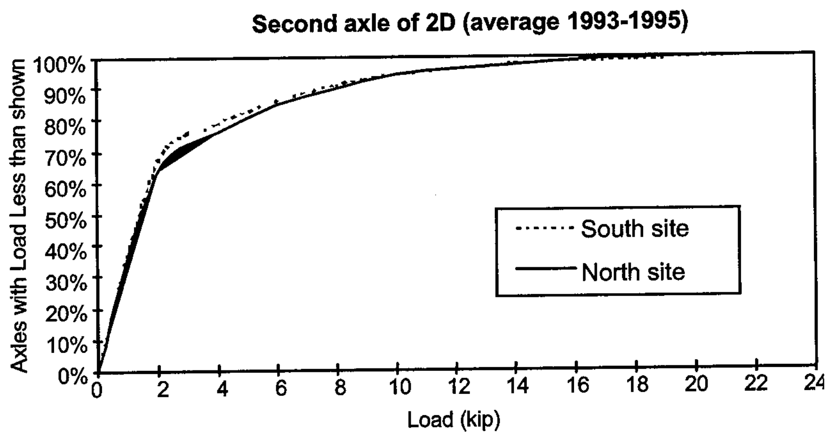


Figure 4.5 Load frequency distribution of the second axle of 2D at north and south sites

4.4 COMPARISON OF LOAD DISTRIBUTION OF SAME AXLE GROUP IN SAME OR DIFFERENT TRUCK CLASS

The same axle groups were observed to have different load distributions when they were in different truck classes or at a different position on the same truck class. Thus, it was necessary to calculate the load frequency distribution by truck class and by the position on the truck.

4.4.1 Difference of Load Distribution of First and Second Tandem Axle on 3S2s

A similar load distribution shape was observed for the two tandem axles on 3S2s. However, the load distribution of the first (tractor) tandem axle was shifted to the right of the second (trailer) tandem axle (see Figure 4.6). This difference indicates that the tractor tandem axle carried a heavier load than the trailer tandem axle.

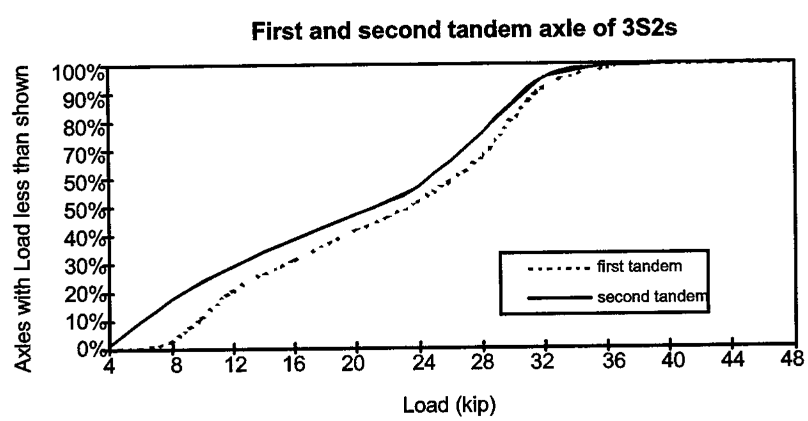


Figure 4.6 Load frequency distribution between the first and second tandem axle of 3S2s

4.4.2 Difference of Load Distribution of the Tractor Tandem Axle Between 3S2 and 3S2 Spread

Different distribution shapes were observed for the tractor tandem axle of 3S2 and 3S2 spread (see Figure 4.7). The two distributions intersected at 28 kip. For loads less than 28 kip, the tractor tandem of a 3S2 spread carried a heavier load than that of a 3S2. For loads more than 28 kip, the tractor tandem of a 3S2 spread carried lighter loads, as compared with a 3S2. The percentage of tractor tandem axles on 3S2s exceeding the legal axle load limit was greater than that of 3S2 spreads.

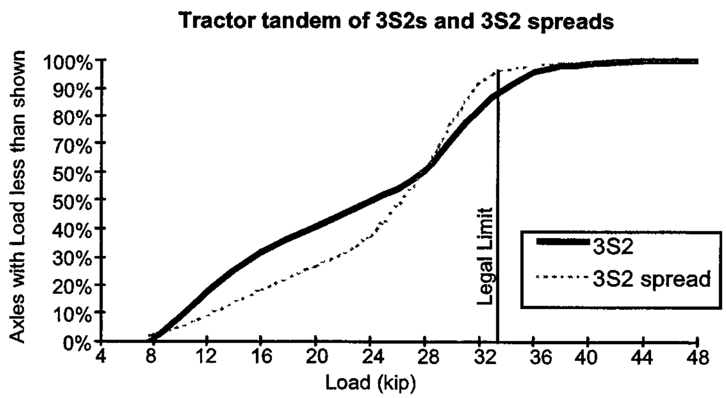


Figure 4.7 Load frequency distribution of the first tandem between 3S2s and 3S2 spreads

4.4.3 Difference of Load Distribution of the Four Single Axles on the 2S1-2

Load frequency distributions of the four single axles on the 2S1-2 are shown in Figure 4.8. The first single (drive) axle carried a heavier load than the second (semi-trailer) single axle. The third axle and the fourth axle (full trailer) have similar load distributions, and both carried less load than the second axle.

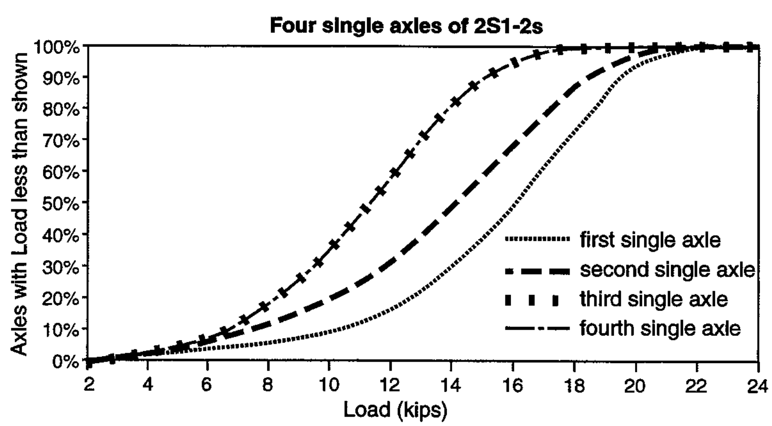


Figure 4.8 Load frequency distribution by four single axles of 2S1-2s

4.4.4 Steering Axle

The percentage of the steering axles over 12 kip was calculated for each truck class. The results of 1995 data are shown in Table 4.1. A small percentage of the steering axles over 12 kip was observed for the large samples. However, taken together the steering axle loads greater than 12 kip may cause additional pavement damage that is not accounted for in current practice.

Table 4.2 Percent of steering axles over 12 kip

Truck Classes	Number of Observations	Percentage over 12 kip
2D	167502	0.15%
3A	30661	4.11%
Rig	977	57.86%
2S1	19663	4.90%
2S2	10178	0.13%
3S1	23606	10.36%
3S2	278395	1.60%
3S3	8891	18.55%
2S1-2	15488	2.23%
2S2-2	159	25.79%
3S1-2	2613	2.03%
3S2-2	458	20.61%

CHAPTER 5. FORECASTING ESALS

For pavement evaluation and design, an estimate of the cumulative traffic loading expected during the analysis period is absolutely essential. The final step, therefore, in using weigh-in-motion (WIM) data as a basis for forecasting future traffic loads is to convert them to a common denominator in terms of expected damage to pavements, the observed loads of various magnitudes carried by steering, single, tandem, and tridem axles that occur in typical arrangements on different types of vehicles (sometimes referred to as mixed traffic). A method, now used widely, for expressing axle loads from mixed traffic vehicles in terms of the equivalent number of passes of a standard axle load was developed in the late 1950s at the AASHO Road Test.

5.1 THE AASHO ROAD TEST AND EQUIVALENT SINGLE AXLE LOAD (ESALS)

The AASHO (now AASHTO) Road Test was completed in 1960 after some two years of testing pavements under controlled traffic loading (Refs. 11 and 12). Axle load equivalency factors for pavement design were derived from a statistical analysis of the AASHO Road Test data. These equivalency factors can be used to convert one application of an axle load of a particular magnitude to the number of applications of a standard axle load (e.g., an 18-kip single axle load), which can be expected to cause equivalent damage to a given pavement structure. The cumulative number of standard (e.g., 18-kip single) axle loads is referred to as the number of equivalent single axle loads (ESALs). The concept and application of axle load equivalent factors is summarized below.

Load Equivalence Factors for Flexible Pavements. Equivalency factor equations for flexible pavements presented in ASTM E1318-94 (Ref 3) are:

$$\log W_t = 5.93 + 9.36\log(\overline{SN} + 1) - 4.79\log(L_1 + L_2) + 4.331\log L_2 + G_t/\beta$$

$$\beta = 0.40 + \frac{0.081(L_1 + L_2)^{3.23}}{(\overline{SN} + 1)^{5.19}\, L_2^{3.23}}$$

where

W_t = number of axle load applications at the end of time t for axle sets with dual tires,

$\overline{SN}$ = structural number, an index derived from an analysis of traffic, roadbed soil conditions, and regional factor that may be converted into the thickness of flexible pavement layers through the use of suitable layer coefficients that are related to the type of material being used in each layer of the pavement structure,

L_1 = load on one single axle, or on one tandem-axle set for dual tires, kip,

L_2 = axle code (one for single axle, two for tandem-axle sets),

p_t = serviceability at end of time t (serviceability is the ability of a pavement at the time of observation to serve high-speed, high-volume automobile and truck traffic.),

G_t = a function (the logarithm) of the ratio of loss in serviceability at time t to the potential loss taken to a point where $p_t = 1.5$, or

$$G_t = \log\left[\frac{4.2 - P_t}{4.2 - 1.5}\right], \text{ and}$$

β = a function of design and load variables that influences the shape of the P-verses-W serviceability curve.

The following equation expresses any axle load (W_t) in terms of the number of applications of the standard 18-kip single-axle load (W_{18}) that would cause the same pavement damage as the (W_t) axle load,

$$E_i = \frac{W_{t_{18}}}{W_{t_i}} = \left[\frac{(L_1 + L_2)^{4.79}}{(18+1)^{4.79}}\right]\left[\frac{10^{G_t/\beta_{18}}}{(10^{G_t/\beta_i})L_2^{4.331}}\right]$$

This expression is defined as the equivalency factor for flexible pavements.

Load Equivalence Factors for Rigid Pavements. Equivalency factor equations for rigid pavements presented in ASTM E1318-94 are:

$$\log W_t = 5.85 + 7.35(\log D + 1) - 4.62\log(L_1 + L_2) + 3.28\log L_2 + G_t/\beta$$

$$\beta = 1.0 + \frac{3.63(L_1 + L_2)^{5.20}}{(D+1)^{8.46} L_2^{3.52}}$$

where

D = thickness of rigid pavement slab, in., and

$$G_t = \log\left[\frac{4.5 - P_t}{4.5 - 1.5}\right].$$

All other terms are as defined above.

The following equation expresses any axle load (W_t) in terms of the number of applications of the standard 18-kip single-axle load (W_{18}) that would cause pavement damage equivalent to that caused by one single pass of the (W_t) axle load,

$$E_i = \frac{W_{t18}}{W_{ti}} = \left[\frac{(L_1 + L_2)^{4.62}}{(18+1)^{4.62}}\right]\left[\frac{10^{G_t/\beta_{18}}}{(10^{G_t/\beta_i})L_2^{\ 3.28}}\right]$$

This expression is defined as the equivalency factor for rigid pavements.

5.1.1 Steering-Axle and Tridem-Axle Equivalency Factors

Test trucks at the AASHO Road Test had steering-axle loads (on single tires) that ranged between 2 and 12 kip, while single-axle loads (on dual tires) were between 2 and 30 kip and tandem-axle loads (on dual tires) were between 24 and 48 kip. In developing the AASHO axle-load equivalency factor equations, the effect of steering-axle loads (less than 12 kip) was not accounted for separately, but was included with that of the other axles on the test vehicle. When analyzing WIM data, it is important, therefore, to include the additional damaging effect of steering-axle loads in excess of 12 kip in ESAL calculations. A table of equivalency factors for steering-axle loads greater than 12 kip on flexible pavements was developed by Carmichael et al. (Ref 6) using a concept of pavement surface curvature and the resulting tensile strains in the asphalt pavement. Their analysis indicated that single-tire loads produced more damage than the same loads on dual tires. This concept has been substantiated by the theoretical work of others (Ref 7). The tabular values of equivalency factors for steering-axle loads greater than 12 kip on flexible pavements that were developed by Carmichael et al. are incorporated into the computer programs described herein.

The analysis techniques used by Carmichael et al. to separate the relative damage to rigid pavements by single and dual-tire loads were considered by the authors to be unsuccessful in developing equivalency factors for steering axles on rigid pavements. As no other work on this problem is known, no additional damage to rigid pavements by steering-axle loads greater than 12 kip is assessed by the computer programs described herein.

Although tridem (triple) axles were not included on the test trucks at the AASHO Road Test, they are now used rather frequently. Carmichael et al. applied the same concept of pavement surface curvature and tensile strain that was used successfully for steering axles to develop equivalency factors for tridems on flexible pavements. Subsequently, such equivalency factors were calculated by Izadmehr (Refs 13, 15, and 16) by setting the term L_2 = 3 in the AASHO flexible pavement equivalency factor equations; in those calculations, there was very close agreement with those shown by Carmichael et al. for flexible pavements. Izadmehr also calculated rigid pavement equivalency factors for tridem axles by the same technique and pointed out that they "appear to be reasonable, but they have not been validated through experimental work" (Ref 15). In the computer programs described herein, tridem-axle load equivalency factors are calculated by setting the term L_2 = 3 in the AASHO equivalency factor equations for both flexible and rigid pavements.

5.1.2 Weighted Average Equivalency Factor for Each Vehicle Type

The weighted average equivalency factor for each vehicle type can be obtained by multiplying the axle load frequency distributions of each vehicle type by their corresponding equivalency factors. An example of the calculation of weighted average equivalency factors for 3S2 and 3S2 spread trucks is shown in Table 5.1. Weighted average equivalency factors for each truck type in the TxDOT classification scheme are calculated in the computer program developed for this study by the same method illustrated in the Table 5.1 for 3S2 and 3S2 spread trucks and by inputting the pavement type, terminal serviceability (p_t), and the structural number (SN) for flexible pavement or the slab thickness (d) for rigid pavement. The weighted average ESALs of each truck classes for 1995 south site data are summarized in Table 5.2.

5.2 DIFFERENCE IN ESALS OF 3S2 AND 3S2 SPREAD

As stated in Section 4.1, the axle group arrangement of a 3S2 spread type truck is different from that of a 3S2. The weighted average ESALs per 3S2 spread and 3S2 truck were calculated to illustrate the difference in their corresponding ESALs. The following assumed data were used for this analysis: flexible pavement, a terminal serviceability $p_t = 2.5$ and a structural number SN = 6. The axle load frequency distributions were obtained for the data of 1995, south site. According to the results of Section 4.2, the axle load frequency distributions of 1995 were almost the same as those of 1993 and 1994. The weighted average ESALs from a 3S2 spread and a 3S2 are shown in Table 5.1. The difference of ESALs per vehicle is 0.82, which is about 60 percent of the ESALs of one 3S2 spread. Pangburn reported (Ref 18) that the ESALs of 3S2 spread account for 12 percent of the daily ESALs by using the same data set as this study. Therefore, by not classifying 3S2 spread trucks separately, about 7 percent of the daily ESALs are not accounted for if these classes are not separated. This can result in significant and unexpected damage to a pavement structure.

5.3 AN EXAMPLE OF ESAL CALCULATIONS FOR FORECASTED YEAR

The following data are needed to forecast ESALs for a future year:

1. *AADT by truck types of base year.* Since traffic growth rate is analyzed individually for each truck type, the AADT by truck type is used in the forecast.
2. *Traffic growth rate assigned to each truck type.* The time series model developed in this study or other applicable methods can be used to obtained the annual growth rate (g). The future cumulative traffic volume can be obtained by multiplying the AADT of the base year by 365 days, and by the following simple growth rate or the compound growth factor, which is used in the AASHTO Guide:

   Simple Growth Rate:

   $$\text{Growth Factor} = n + \frac{1}{2} n\,(n-1)\,g$$

Compound Growth Rate:

$$\text{Growth Factor} = \frac{[(1+g)^n - 1]}{g}$$

where:

g = annual growth rate

n = analysis period in years

3. *Axle weight distribution of each truck type for the forecasted year.* According to the analysis described in Chapter 4, the axle weight distribution remained virtually constant over the analysis period.
4. *Weighted ESALs per vehicle by truck type.* Weighted ESALS per vehicle can be calculated by multiplying the axle weight distributions by their respective equivalency factors. Pavement type, terminal serviceability (p_t), and pavement structural number (SN) or slab thickness (d), are needed to obtain the equivalency factors

A computer program was developed for this study to automatically calculate the cumulative ESALs for the analysis period. An example of the output of the program is shown in Table 5.3. In this example, a 20-year analysis period is used, and a flexible pavement with pt = 2.5 and SN = 6 is assumed. The data for AADT and weighted ESALs per vehicle were obtained from 1995 WIM data at the site south of Corrigan. The number of ESALs derived using this procedure represents the total ESALs for two lanes. This number can be distributed between lanes using a lanewise distribution factor.

Table 5.1 Calculation of ESALs for 3S2 and 3S2 spread (data from 1995, south site)

				3S2				3S2 spread					
Single axle		Tandem axle		First Tandem		Second Tandem		Tandem		First Single		Second Single	
Weight range	Equiv. factor	Weight range	Equiv. factor	Weight Dist.	ESALs	Weight Dist.	ESALs	Weight Dist.	ESALs	Weight Dist.	ESALs	Weight Dist.	ESALs
0-2	0.0002	0-4	0	0.2%	0.0000	1.2%	0.0000	0.7%	0.0000	3.3%	0.0000	5.2%	0.0000
2-4	0.002	4-8	0.001	2.7%	0.0000	17.3%	0.0002	2.1%	0.0000	3.7%	0.0001	4.3%	0.0001
4-6	0.009	8-12	0.006	19.2%	0.0012	11.9%	0.0007	9.9%	0.0006	7.9%	0.0007	7.2%	0.0007
6-8	0.031	12-16	0.024	10.7%	0.0026	9.4%	0.0023	8.4%	0.0020	5.5%	0.0017	5.8%	0.0018
8-10	0.08	16-20	0.07	10.3%	0.0072	8.5%	0.0060	9.0%	0.0063	6.4%	0.0051	6.4%	0.0051
10-12	0.176	20-24	0.166	9.7%	0.0161	9.6%	0.0159	13.2%	0.0220	6.0%	0.0105	6.0%	0.0106
12-14	0.342	24-28	0.342	16.7%	0.0573	20.3%	0.0694	32.7%	0.1117	10.0%	0.0342	11.1%	0.0380
14-16	0.606	28-32	0.633	23.8%	0.1507	18.3%	0.1160	22.1%	0.1398	25.0%	0.1518	28.6%	0.1734
16-18	1	32-36	1.08	5.6%	0.0608	2.7%	0.0296	1.5%	0.0164	25.8%	0.2576	21.9%	0.2189
18-20	1.55	36-40	1.73	0.9%	0.0153	0.6%	0.0100	0.3%	0.0046	5.4%	0.0836	2.9%	0.0456
20-22	2.3	40-44	2.61	0.1%	0.0019	0.1%	0.0036	0.1%	0.0037	0.8%	0.0192	0.5%	0.0112
22-24	3.27	44-48	3.79	0.0%	0.0000	0.0%	0.0000	0.0%	0.0000	0.2%	0.0055	0.1%	0.0020
Weighted Average ESALs per axle					0.3130		0.2536		0.3072		0.5700		0.5073
Total ESALs per Truck Type				0.5666				1.3845					

Table 5.2 Summary of weighted average ESALs for each truck class

Truck classes	Weighted Average ESAL Value
2D	0.0311
2D-1 axle trailer	0.1531
2D-2 axle trailer	0.351
3-axle single unit (3A)	0.1780
4-axle single unit (4A)	0.2969
4-axle single unit (Rig)	0.7768
2S1	0.2270
2S2	0.2613
3S1	0.1467
2S3	0.1109
3S2	0.5666
3S2 spread	1.3845
3S3	0.8509
2S1-2	1.5206
2S2-2	0.5001
3S1-2	0.6154
3S2-2	0.7452

Table 5.3 Example of calculation of ESALs (compound growth rate)

					$p_t = 2.5$
Analysis Period: 20 years					SN = 6
Truck Types	Current Traffic (AADT)	Growth rate	Design Traffic (20 years' accumulation)	Weighted ESALs (/vehicle)	Design ESALs
Type 4	600	5%	7,240,140	0.031	224,444
Type 5	70	0%	511,000	0.178	90,958
Type 6	2	0%	14,600	0.297	4,336
Type 7	30	0%	219,000	0.277	60,663
Type 8	100	0%	730,000	0.261	190,530
Type 9	1,100	6%	14,771,185	0.567	8,375,261
Type 10	20	0%	146,000	0.851	124,246
Type 11	60	0%	438,000	1.521	666,198
Type 12	10	0%	73,000	0.615	44,895
Type 13	1	0%	7,300	0.745	5,438
All Truck	1,993		24,150,225	Total ESALs	9,746,969

CHAPTER 6. CONCLUSIONS AND RECOMMENDATIONS

The work reported herein is part of a continuing research project that is concerned with the implementation of a long-range pavement rehabilitation plan in TxDOT's Lufkin District on US 59 in east Texas. The focus of this work has been on the development of an axle-load forecasting methodology, one based on the processing of voluminous traffic data (about 7,000 vehicle records per day for 41 months at two stations) that have been obtained from special weigh-in-motion (WIM) systems located within two pavement test sites near Corrigan, Texas. The principal objectives were (1) to develop efficient computer code for processing the binary data files recorded by the WIM systems and for automatically grouping vehicles according to TxDOT classes; (2) to process the WIM data to develop axle-load frequency distributions for the individual axles on each vehicle class; (3) to identify an effective axle-load forecasting procedure for each vehicle class; and (4) to implement the computer code needed to estimate the cumulative equivalent single axle loads (ESALs) that can be expected over a selected future time period. Accomplishment of these objectives serves as the basis for the following conclusions and recommendations.

6.1 CONCLUSIONS

1. The C-language computer program, which was written to address the first two objectives, processes one month's worth of the binary data files recorded by a WIM system in about 10 minutes on a Pentium-processor PC. The computer program that was developed previously to accomplish these functions used a Microsoft EXCEL macro, which was recognized as being inefficient, but which proved valuable in the early stages of the project to establish a strategy for analyzing the complex data sets. The EXCEL macro program required approximately 8 hours on the same computer to accomplish similar results. Thus, the new C-language program is much more efficient.
2. The overall error rate in classifying vehicles into TxDOT classes (via number and spacing of axles on the vehicle) was about 0.5 percent of all vehicles observed during 1995 at the south WIM site. Small adjustments were made to the axle-spacing values currently used by TxDOT in order to include certain vehicles in the proper class. Possible causes of the erroneous classification are discussed.
3. Of all vehicles, motorcycles, 2-axle passenger cars, pickup trucks and buses accounted for 73 percent, and trucks accounted for 27 percent. Of the trucks, 59 percent were 5-axle semi-trailers, and 26 percent were 2-axle 6-tire single units.
4. Traffic counts between the north and south WIM sites were almost identical on the weekend (Saturday and Sunday). During weekdays, the traffic count at the north site was always slightly higher than that at the south site.
5. In general, 72 percent of the vehicles traveled in the right-hand lane and 28 percent traveled in the left-hand lane. Moreover, 80 percent of the trucks traveled in the right-hand lane.

6. A linear regression method was used to evaluate the annual change (growth rate) in the number of 5-axle semi-trailer trucks that used US 59 near the WIM station on an average day, as reflected in TxDOT reports for 1986 to 1994 (1987 and 1989 data were not available). The reported values were derived from one 24-hour, non-directional manual classification count per year, adjusted to reflect an average daily count at two locations on US 59 adjacent to (one north of and one south of) its intersection with FM 62 in Polk County, and only a few miles south of the WIM stations. The annual, non-directional growth rate for this truck class during these years was approximately 4 percent.
7. A time-series method was used with data obtained from the south-site WIM system to study changes in the count of southbound 5-axle semi-axle trailer trucks during a 41-month period beginning in January 1993. The annual growth rate for 5-axle semi-trailer trucks is about 7 percent from the WIM data, which were sampled from almost every vehicle passing by the WIM site. The WIM data have a larger sample size but a shorter time span than the TxDOT data.
8. There was no change of loading pattern during 1993, 1994, and 1995. The axle weight distribution for the same axle group remained the same. It was observed that the axle weight distributions for the same axle were similar or almost identical between the two WIM sites. The same axle groups were observed to have different load distributions when they were in different truck classes or at a different position on the same truck class. Thus, it is necessary to calculate the load frequency distribution by truck class and by the axle position on the truck.
9. A small percentage of steering axles over 12 kip was observed in the large samples. These steering-axle loads greater than 12 kip may cause additional pavement damage that is not usually accounted for properly in current practice. Their effects are, however, included in the analyses (computer programs) reported herein.
10. The 3S2-spread truck was found to have axle loads heavier than those of the normal 3S2 truck. Moreover, the ESALs generated by the observed 3S2-spreads were over 2 times more than those generated by the 3S2 trucks.

6.2 RECOMMENDATIONS

1. A study of economic factors, the population, and lane-use of the studied area will help to improve the forecast of future traffic.
2. Traffic growth rate and a seasonal factor for 2D trucks can be obtained by making another time-series analysis similar to the one described for 5-axle, semi-trailer trucks.
3. Based on evaluation of the TxDOT classification scheme, it might be desirable to separate some of the axle groups. Trucks, such as 3S2 and 2S3, whose axle weight distributions are very different, should be classified separately.
4. It is necessary to add WIM system capability on northbound US 59 close to the existing sites to quantify the traffic count and loading data for this direction.

REFERENCES

1. American Association of State Highway and Transportation Officials, "AASHTO Guidelines for Traffic Data Program," Joint Task Force on Traffic Monitoring Standards of the AASHTO Highway Subcommittee on Traffic Engineering, Washington, D.C., 1992.
2. American Association of State Highway and Transportation Officials, "AASHTO Guide for Design of Pavement Structures," Washington, D.C., 1993.
3. American Society for Testing and Materials, "Standard Specification for Highway Weigh-in-Motion (WIM) Systems with User Requirements and Test Method," Designation E 1318-94, Reprinted from the Annual Book of ASTM Standards, Section 4—Construction, Vol. 04.03, Philadelphia, 1995.
4. Ang, A. H-S., and Tang, W. H., *Probability Concepts in Engineering Planning and Design*, John Wiley and Sons, 1975.
5. Bowerman, B. L., and O'Connell, R. T., *Time Series Forecasting: Unified Concepts And Computer Implementation*, 2nd edition, Duxbury Press, Boston, 1979.
6. Carmichael, III, R. F., Roberts, F. L., Jodahi, P. R., Treybig, H. J., and Finn, F. N., "Effects of Changes in Legal Load Limits on Pavement Costs: Volume 1 — Development of Evaluation Procedure," Federal Highway Administration Report No. FHWA-RD-78-98, July 1978.
7. Deacon, J. A., "Load Equivalency in Flexible Pavements," *Proceedings*, Association of Asphalt Paving Technologists, Vol. 38, 1969.
8. Deitel, H. M., and Deitel, P. J., *C: How To Program*, 2nd edition, Prentice Hall International, Inc., 1994.
9. Garner, J. E., and Lee, C. E., "Augmented Weigh-in-Motion System for Traffic Data Collection and Analysis," Transportation Research Record No. 1501, Pavement-Vehicle Interaction and Traffic Monitoring, Transportation Research Board, National Research Council, National Academy Press, Washington, D.C., 1995, pp. 81-86.
10. Gujarati, D. N., *Basic Econometrics,* 3rd edition, McGraw-Hill, Inc., 1995.
11. Highway Research Board, "The AASHO Road Test," Report 5, Pavement Research, Highway Research Board Special Report 61E, 1962.
12. Highway Research Board, "The AASHO Road Test," Proceedings of a conference held May 16-18, 1962, St. Louis, Missouri, Special Report 73, Washington, D.C., 1974.
13. Izadmehr, B., "Truck Traffic Loading on Multilane Highways," master's thesis, The University of Texas at Austin, 1982.
14. Janacek, G., and Swift, L., *Time Series: Forecasting, Simulation, Applications*, Eillis Horwood, 1993.

15. Lee, C. E., Izadmehr, B., Machemehl, R. B., "Demonstration of Weigh-in-Motion Systems for Data Collection and Enforcement," Report No. 557-1F, Center for Transportation Research, The University of Texas at Austin, 1985.

16. Lee, C. E., Shankar, P. R., and Izadmehr, B., "Lateral Placement of Trucks in Highway Lanes," Report No. 310-1F, Center for Transportation Research, The University of Texas at Austin, 1983.

17. Leidy, J. P., "Measurement and Analysis of Traffic Loads Along the Texas-Mexico Border," master's thesis, The University of Texas at Austin, 1994.

18. Pangburn, J. W., "Preliminary Research Findings on Traffic-Load Forecasting Using Weigh-in-Motion Data," Research Report 987-5, Center for Transportation Research, The University of Texas at Austin, April 1996.

19. PAT Equipment Corporation, Inc., "DAW 100 – Bending Plate, System Description of Automatic Vehicle Weight and Classification System," 1992.

20. Texas Department of Transportation, "Manual Recorder Instruction Manual," Transportation Planning and Programming Division, 1996.

21. U.S. Department of Transportation, Federal Highway Administration, "Traffic Monitoring Guide," Office of Highway Information Management, Washington, D.C., 1995.

APPENDIX A.

A.1 CAR PROGRAM
COMPUTER PROGRAM FOR VEHICLE SORTING AND AXLE-LOAD FREQUENCY DISTRIBUTION

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <dos.h>
#include <math.h>
#define MAP1 12
#define MAP2 12
#define MAP4 17
#define MAP3 4
#define OP_COUNT 1
#define OP_WEIGHT 2

int Calculate(char *infile, char *outfile);
void FindWeight(int lane, int length, double * data);
static int weight[13][5];
static int map1(double data);
static int map2(double data);
static int map3(double data);
static void Initialize();
static void PrintWeight(char *ff, char *outfile);
static long get_sum(long *tt, int num, double *ss);
static double get_beta(double L1, double L2);
static double get_factor(double L1, double L2);
static double get_wt(double L1, double L2);

// type 4
static long     s41[MAP1], s42[MAP1], s43[MAP1], s44[MAP1];
static long     t42[MAP2];
static long     e41[MAP3], e42[MAP3], e43[MAP3];
// type 5
static long     t5[MAP2], e5[MAP3];
// type 6
static long     s6[MAP1];
static long     t6[MAP2], r6[MAP4];
static long     e61[MAP3], e62[MAP3];
// type 7
static long s71[MAP1], s72[MAP1];
static long e7[MAP3];
// type 8
static long     s81[MAP1], s82[MAP1];
static long     t81[MAP2], t82[MAP2];
static long     e81[MAP3], e82[MAP3];
// type 9
static long s9[MAP1], ds91[MAP1], ds92[MAP1];
static long r9[MAP4], t91[MAP2], t92[MAP2], dt9[MAP2];
```

```
static long e91[MAP3], e92[MAP3], e93[MAP3];
// type 10
static long     r10[MAP4], t10[MAP2];
static long     e10[MAP3];
// type 11
static long s111[MAP1], s112[MAP1], s113[MAP1], s114[MAP1];
static long e11[MAP3];
// type 12
static long     s1211[MAP1],s1212[MAP1],s1213[MAP1],
                    s1221[MAP1],s1222[MAP1],s1223[MAP1];
static long     t121[MAP2], t122[MAP2];
static long     e121[MAP3],e122[MAP3];
// type 13
static long     t131[MAP2],t132[MAP2];
static long     s131[MAP1],s132[MAP1];
static long     e13[MAP3];

static int op;
static int pave;   /* pavement */
static double serv; /* terminal serviceability */
static double sn;     /* structural number */
static double gt;
static double beta18;
static double wt18;
static double factor1[MAP1], factor2[MAP2], factor3[MAP4];

static double   ffs41 = 0.0, ffs42 = 0.0, ffs43 = 0.0, ffs44 = 0.0;
static double   fft42 = 0.0;
static double   fft5 = 0.0;
static double   ffs6 = 0.0, fft6 = 0.0, ffr6 = 0.0;
static double   ffs71 = 0.0, ffs72 = 0.0;
static double   ffs81 = 0.0, ffs82 = 0.0;
static double   fft81 = 0.0, fft82 = 0.0;
static double   ffs9 = 0.0, ffds91 = 0.0, ffds92 = 0.0;
static double   ffr9 = 0.0, fft91 = 0.0, fft92 = 0.0, ffdt9 = 0.0;
static double   ffr10 = 0.0, fft10 = 0.0;
static double   ffs111 = 0.0, ffs112 = 0.0, ffs113 = 0.0, ffs114 = 0.0;
static double   ffs1211 = 0.0, ffs1212 = 0.0, ffs1213 = 0.0,
                        ffs1221 = 0.0, ffs1222 = 0.0, ffs1223 = 0.0;
static double   fft121 = 0.0, fft122 = 0.0;
static double   fft131 = 0.0, fft132 = 0.0;
static double   ffs131 = 0.0, ffs132 = 0.0;

main()
{
```

```
	char	yy[5], mm[5], site[5];
	char	ff[100], infile[100], outfile[100];
	struct _find_t  c_file;

	//	build the string ff="v00xmm*.yy"
	printf("enter the year (e.g. 94): ");
	scanf("%s", yy);
	printf("enter the month (e.g. 03): ");
	scanf("%s", mm);
	printf("enter the site (1 or 2): ");
	scanf("%s", site);
//	printf("enter the input file: ");
//	scanf("%s", infile);
	sprintf(ff, "v00%s%s*.%s", site, mm, yy);

	printf("enter the output file: ");
	scanf("%s", outfile);

	printf("1: count, 2: weight.: ");
	scanf("%d", &op);

	if (op == OP_WEIGHT) {
		printf("1: flexible pavement, 2: rigid pavement: ");
		scanf("%d", &pave);
		if (pave != 1 && pave != 2) {
			printf("please enter 1 or 2 for pavement\n");
		}

	printf("terminal serviceability (0.0 - 5.0): ");
	scanf("%lf", &serv);

	if (pave == 1) {
		printf("structural number (1 - 9): ");
	} else {
		printf("slab thinkness (6 - 14): ");
	}
	scanf("%lf", &sn);

	if (pave == 1)
		gt = log10((4.2 - serv) / 2.7);
	else
		gt = log10((4.5 - serv) / 3.0);
    } // end if for weight

	Initialize();
```

```
    if (_dos_findfirst(ff, _A_NORMAL, &c_file) != 0) {
      exit(0);
    }
    strcpy(infile, c_file.name);
        if (Calculate(infile, outfile) == 0) {
                exit(0);
        }
        while (_dos_findnext(&c_file) == 0) {
                strcpy(infile, c_file.name);
                if (Calculate(infile, outfile) == 0) {
                        break;
                }
        }
        if (op == OP_WEIGHT) {
                PrintWeight(ff, outfile);
        }
        return 0;
}


int Calculate(char *infile, char *outfile)
{
        char    str[500], *p, *pp, str2[500];
        FILE    *f, *fw;
        double data[50];
        int             i, type, length, lane;
        long    count1[20], count2[20], count = 0, sum1 = 0, sum2 = 0, line_no=0;

        f = fopen(infile, "rt");
        fw = fopen("error.txt","wt");
        if (f == NULL) {
                printf("Cannot open input file - %s\n", infile);
                return 0;
        }

        for (i = 0; i < 20; i++)
                count1[i] = count2[i] = 0;
//      while (fscanf(f, "%s", str) != EOF) { //        read line by line from file
        while (fgets(str, 250, f) != NULL || !feof(f)) {        //      read line by line from file
                strcpy(str2, str);
                if (strlen(str) > 240)
                        goto error;
                line_no++;
                pp = str;
```

```
length = 0;
while (*pp && length <= 35) {        //        decode data from string
        p = pp;
        do {
                pp++;
        } while ((*pp != 0) && (*pp != ','));
        if (*pp == ',') {
                *pp = 0;
                pp++;
        }
        length++;
        if (length == 1) {
                lane = atoi(p);
                if ((lane != 1 ) && (lane != 2))
                        break;
        } else if (((length == 8)||(length == 9)||(length == 12)||(length == 16)
||(length == 20)||(length == 24)||(length ==28)) && (*p == '%')) {
                data[length] = -1;
                continue;
        } else {
                data[length] = atof(p);
                if (!sscanf(p, "%lf", &(data[length]))) {
                        length = 0;
                        break;
                }
        }
}
type = 14;
switch (length) {
case 35:
case 31:
        if ((data[length] >= 11.1) && (data[length] <= 24.0))
                type = 13;
        else if ((data[length] >= 3.4) && (data[length] <= 6.0))
                type = 10;
        if ((length == 31) && (type == 13))
                type = 12;
        break;
case 27:
        if ((data[27] >= 2.5) && (data[27] <= 6.0))
                type = 9;
        else if ((data[27] >= 6.1) && (data[27] <= 40.0)) {
                if ((data[19] >= 8.0) && (data[19] <= 50.0))
                        type = 11;
                else if ((data[19] >= 2.5) && (data[19] <= 8.0))
```

```
                    type = 9;
        }
        break;
case 23:
        if ((data[23] >= 6.1) && (data[23] <= 40.0))
                type = 8;
        else if ((data[23] >= 3.4) && (data[23] <= 6.0)) {
                if ((data[19] >= 3.4) && (data[19] <= 4.7))
                        type = 6;
                else if (data[19] >= 5.0) {
                        if ((data[15] >= 6.1 && data[15] <= 20.0))
                                type = 8;
                        else if ((data[15] >= 0.1) && (data[15] <= 6.0))
                                type = 6;
                }
        } else if ((data[23] >= 0.1) && (data[23] <= 3.3)) {
                if ((data[15] >= 13.1) && (data[15] <= 20.9))
                        type = 4;
                else if ((data[15] >= 10.3) && (data[15] <= 13.0))
                        type = 2;
                else if ((data[15] >= 6.1) && (data[15] <= 10.2))
                        type = 1;
        }
        break;
case 19:
        if ((data[19] >= 20.2) && (data[19] <= 60.0))
                type = 7;
        else if ((data[19] >= 6.1) && (data[19] <= 20.1)) {
                if ((data[15] >= 13.1) && (data[15] <= 20.9))
                        type = 4;
                else if ((data[15] >= 10.3) && (data[15] <= 13.0))
                        type = 2;
                else if ((data[15] >= 6.1) && (data[15] <= 10.2))
                        type = 1;
        } else if ((data[19] >= 3.4) && (data[19] <= 6.0)) {
                if ((data[15] >= 6.1) && (data[15] <= 20.9))
                        type = 5;
                else if ((data[15] >= 21.0) && (data[15] <= 40.0))
                        type = 3;
        }
        break;
case 15:
        type = ((data[15] >= 13.1) && (data[15] <= 20.9))? 4 :
                    ((data[15] >= 21.0) && (data[15] <= 40.0))? 3 :
                    ((data[15] >= 10.3) && (data[15] <= 13.0))? 2 :
```

```
                                        ((data[15] >= 0.1) && (data[15] <= 10.2))? 1 : 14;
                        break;
                }       //      switch (length)
                FindWeight(type, length, data);
                if (lane == 1)
                        count1[type]++;
                else if (lane == 2)
                        count2[type]++;
                if (type == 14) {
error:
                        fprintf(fw, "line %ld: %s\n", line_no, str2);
                }
                count++;
                if (!(count % 100))
                        printf("%ld\n", count);
        }       //      while()
        fclose(f);
  fclose(fw);

        if (op == OP_COUNT) {
                f = fopen(outfile, "at");
                if (f == NULL) {
                        printf("Cannot open output file - %s\n", outfile);
                        return 0;
                }
                fprintf(f, "Traffic count on %d/%d\n", (int)(data[2]+0.5), (int)(data[3]+0.5));
                fprintf(f, "type    lane 1    lane 2\n");
                for (i = 1; i <= 14; i++) {
                fprintf(f, " %2d    %6ld    %6ld\n", i, count1[i], count2[i]);
                        sum1 = sum1 + count1[i];
                        sum2 = sum2 + count2[i];
                }
                fprintf(f, "Total  %6ld    %6ld\n", sum1, sum2);
                fclose(f);
        }
        return 1;
}

void FindWeight(int type, int length, double * data)
{
        double tt;
        int    td;

        switch (type) {
```

```
case 4:
        if (length == 15) {
                tt = data[13]+data[14];
                td = map1(tt);
                if (td >= 0) {
                        s41[td]++;
                }
                tt = data[10]+data[11];
                if (tt > 12 && tt <= 20) {
                        td = (int)ceil((tt - 14)/2);
                        e41[td]++;
                }
        } else if (length == 19) {
                tt = data[13]+data[14];
                td = map1(tt);
                if (td >= 0) {
                        s42[td]++;
                }
                tt = data[17]+data[18];
                td = map1(tt);
                if (td >= 0) {
                        s43[td]++;
                }
                tt = data[10]+data[11];
                if (tt > 12 && tt <= 20) {
                        td = (int)ceil((tt - 14)/2);
                        e42[td]++;
                }
        } else if (length == 23) {
                tt = data[17]+data[18]+data[21]+data[22];
                td = map2(tt);
                if (td >= 0) {
                        t42[td]++;
                }
                tt = data[13]+data[14];
                td = map1(tt);
                if (td >= 0) {
                        s44[td]++;
                }
                tt = data[10]+data[11];
                if (tt > 12 && tt <= 20) {
                        td = (int)ceil((tt - 14)/2);
                        e43[td]++;
                }
        }
```

```
        break;
case 5:
        tt = data[13]+data[14]+data[17]+data[18];
        td = map2(tt);
        if (td >= 0) {
                t5[td]++;
        }

        tt = data[10]+data[11];
        if (tt > 12 && tt <= 20) {
                td = (int)ceil((tt - 14)/2);
                e5[td]++;
        }
        break;
case 6:
        if (data[19] > 6) {
                tt = data[17]+data[18]+data[21]+data[22];
                td = map2(tt);
                if (td >= 0) {
                        t6[td]++;
                }

                tt = data[13]+data[14];
                td = map1(tt);
                if (td >= 0) {
                        s6[td]++;
                }

                tt = data[10]+data[11];
                if (tt > 12 && tt <= 20) {
                        td = (int)ceil((tt - 14)/2);
                        e61[td]++;
                }
        } else {
                tt = data[13]+data[14]+data[17]+data[18]+data[21]+data[22];
                td = map3(tt);
                if (td >= 0) {
                        r6[td]++;
                }

                tt = data[10]+data[11];
                if (tt > 12 && tt <= 20) {
                        td = (int)ceil((tt - 14)/2);
                        e62[td]++;
                }
```

```
		}
		break;
	case 7:
		tt = data[13]+data[14];
		td = map1(tt);
		if (td >= 0) {
			s71[td]++;
		}
		tt = data[17]+data[18];
		td = map1(tt);
		if (td >= 0) {
			s72[td]++;
		}
		tt = data[10]+data[11];
		if (tt > 12 && tt <= 20) {
			td = (int)ceil((tt - 14)/2);
			e7[td]++;
		}
		break;
	case 8:
		if (data[23] <= 6) {
			tt = data[13]+data[14];
			td = map1(tt);
			if (td >= 0) {
				s81[td]++;
			}
			tt = data[17]+data[18]+data[21]+data[22];
			td = map2(tt);
			if (td >= 0)
				t81[td]++;
			tt = data[10]+data[11];
			if (tt > 12 && tt <= 20) {
				td = (int)ceil((tt - 14)/2);
				e81[td]++;
			}
		} else {
			tt = data[13]+data[14]+data[17]+data[18];
			td = map2(tt);
			if (td >= 0)
				t82[td]++;
			tt = data[21]+data[22];
			td = map1(tt);
			if (td >= 0) {
				s82[td]++;
			}
```

```
                tt = data[10]+data[11];
                if (tt > 12 && tt <= 20) {
                        td = (int)ceil((tt - 14)/2);
                        e82[td]++;
                }
        }
        break;
case 9:
        if (data[23] <= 6.0) {
                tt = data[13] + data[14];
                td = map1(tt);
                if (td >= 0) {
                        s9[td]++;
                }
                tt = data[17]+data[18]+data[21]+data[22]+data[25]+data[26];
                td = map3(tt);
                if (td >= 0)
                        r9[td]++;
                tt = data[10]+data[11];
                if (tt > 12 && tt <= 20) {
                        td = (int)ceil((tt - 14)/2);
                        e91[td]++;
                }
        } else if (data[27] <= 6) {
                tt = data[13] + data[14] + data[17]+data[18];
                td = map2(tt);
                if (td >= 0) {
                        t91[td]++;
                }
                tt = data[21]+data[22]+data[25]+data[26];
                td = map2(tt);
                if (td >= 0) {
                        t92[td]++;
                }
                tt = data[10]+data[11];
                if (tt > 12 && tt <= 20) {
                        td = (int)ceil((tt - 14)/2);
                        e92[td]++;
                }
        } else {
                tt = data[13] + data[14] + data[17]+data[18];
                td = map2(tt);
                if (td >= 0) {
                        dt9[td]++;
                }
```

```
                tt = data[21]+data[22];
                td = map1(tt);
                if (td >= 0) {
                        ds91[td]++;
                }
                tt = data[25]+data[26];
                td = map1(tt);
                if (td >= 0) {
                        ds92[td]++;
                }
                tt = data[10]+data[11];
                if (tt > 12 && tt <= 20) {
                        td = (int)ceil((tt - 14)/2);
                        e93[td]++;
                }
        }
        break;
case 10:
        tt = data[21]+data[22]+data[25]+data[26]+data[29]+data[30];
        td = map3(tt);
        if (td >= 0) {
                r10[td]++;
        }
        tt = data[13]+data[14]+data[17]+data[18];
        td = map2(tt);
        if (td >= 0) {
                t10[td]++;
        }
        tt = data[10]+data[11];
        if (tt > 12 && tt <= 20) {
                td = (int)ceil((tt - 14)/2);
                e10[td]++;
        }
        break;
case 11:
        tt = data[13] + data[14];
        td = map1(tt);
        if (td >= 0) {
                s111[td]++;
        }
        tt = data[17]+data[18];
        td = map1(tt);
        if (td >= 0) {
                s112[td]++;
        }
```

```
        tt = data[21]+data[22];
        td = map1(tt);
        if (td >= 0) {
                s113[td]++;
        }
        tt = data[25]+data[26];
        td = map1(tt);
        if (td >= 0) {
                s114[td]++;
        }
        tt = data[10]+data[11];
        if (tt > 12 && tt <= 20) {
                td = (int)ceil((tt - 14)/2);
                e11[td]++;
        }
        break;
case 12:
        if (data[23] <= 6) {
                tt = data[13]+data[14];
                td = map1(tt);
                if (td >= 0) {
                        s1211[td]++;
                }
                tt = data[17]+data[18]+data[21]+data[22];
                td = map2(tt);
                if (td >= 0)
                        t121[td]++;
                tt = data[25]+data[26];
                td = map1(tt);
                if (td >= 0)
                        s1212[td]++;
                tt = data[29]+data[30];
                td = map1(tt);
                if (td >= 0)
                        s1213[td]++;
                tt = data[10]+data[11];
                if (tt > 12 && tt <= 20) {
                        td = (int)ceil((tt - 14)/2);
                        e121[td]++;
                }
        } else {
                tt = data[13]+data[14]+data[17]+data[18];
                td = map2(tt);
                if (td >= 0)
                        t122[td]++;
```

```
                tt = data[21]+data[22];
                td = map1(tt);
                if (td >= 0) {
                        s1221[td]++;
                }
                tt = data[25]+data[26];
                td = map1(tt);
                if (td >= 0) {
                        s1222[td]++;
                }
                tt = data[29]+data[30];
                td = map1(tt);
                if (td >= 0) {
                        s1223[td]++;
                }
                tt = data[10]+data[11];
                if (tt > 12 && tt <= 20) {
                        td = (int)ceil((tt - 14)/2);
                        e122[td]++;
                }
        }
        break;
case 13:
        tt = data[17]+data[18]+data[13]+data[14];
        td = map2(tt);
        if (td >= 0)
                t131[td]++;

        tt = data[21]+data[22]+data[25]+data[26];
        td = map2(tt);
        if (td >= 0)
                t132[td]++;

        tt = data[29]+data[30];
        td = map1(tt);
        if (td >= 0)
                s131[td]++;

        tt = data[33]+data[34];
        td = map1(tt);
        if (td >= 0)
                s132[td]++;

        tt = data[10]+data[11];
        if (tt > 12 && tt <= 20) {
```

```
                        td = (int)ceil((tt - 14)/2);
                        e13[td]++;
                }

                break;
        }
}

static int map1(double tt)
{
        int td = -1;

        if (tt > 0 && tt <= 3) {
                td = 0;
        } else if (tt > 23) {
                td = 12;
        } else if (tt > 3 && tt <= 23) {
                td = (int)ceil(tt-1)/2;
        }
        return td;
}

static int map2(double tt)
{
        int td = -1;
        if (tt > 0 && tt <= 6) {
                td = 0;
        } else if (tt > 46) {
                td = 12;
        } else if (tt > 6 && tt <= 46) {
                td = (int)ceil((tt - 6)/4);
        }
        return td;
}

static int map3(double tt)
{
        int td = -1;
        if (tt > 0 && tt <= 6) {
                td = 0;
        } else if (tt > 66) {
                td = 17;
        } else if (tt > 6 && tt <= 66) {
                td = (int)ceil((tt - 6)/4);
        }
```

```
	return td;
}

static void Initialize()
{
	int i;

	for (i=0; i < MAP1; i++) {
		s41[i] = s42[i] = s43[i] = s44[i] = 0;
		s6[i] = s71[i] = s72[i] = s81[i] = s82[i] = 0;
		s9[i] = ds91[i] = ds92[i] = s111[i] = 0;
		s112[i] = s113[i] = s114[i] = s131[i] = s132[i] = 0;
		s1211[i] = s1212[i] = s1213[i] = s1221[i] = s1222[i] = s1223[i] = 0;
	}
	for (i = 0; i < MAP2; i++) {
		t42[i] = t5[i] = t6[i] = t81[i] = t82[i] = 0;
		t91[i] = t92[i] = dt9[i] = t10[i] = 0;
		t121[i] = t122[i] = t131[i] = t132[i] = 0;
	}
	for (i = 0; i < MAP3; i++) {
		e41[i] = e42[i] = e43[i] = e5[i] = e61[i] = e62[i] = e7[i] = 0;
		e81[i] = e82[i] = e91[i] = e92[i] = e93[i] = e10[i] = 0;
		e11[i] = e121[i] = e122[i] = e13[i] = 0;
	}
	for (i = 0; i < MAP4; i++) {
		r6[i] = r9[i] = r10[i] = 0;
	}
	beta18 = get_beta(18, 1);
	wt18 = get_wt(18, 1);
	for (i = 0; i < MAP1; i++) {
		factor1[i] = get_factor((double)(i*2+2), 1);
	}
	for (i = 0; i < MAP2; i++) {
		factor2[i] = get_factor((double)(i*4+4), 2);
	}
	for (i = 0; i < MAP4; i++) {
		factor3[i] = get_factor((double)(i*4+4), 3);
	}
}


static void PrintWeight(char *ff, char *outfile)
{
	FILE	*f;
	int		i;
```

```
long    sum1, sum2, sum3, sum4, sum5, sum6;
double  summ1, summ2, summ3, summ4, summ5, summ6;
double  ppp1, ppp2, ppp3, ppp4;

f = fopen(outfile, "wt");
if (f == NULL) {
        printf("Cannot open file %s", outfile);
        return;
}
fprintf(f, "FILE: %s\n", ff);

fprintf(f, "TYPE 4:\n");
sum1 = get_sum(s41, MAP1, &summ1);
sum2 = get_sum(s42, MAP1, &summ2);
sum3 = get_sum(s43, MAP1, &summ3);
sum4 = get_sum(s44, MAP1, &summ4);
fprintf(f, "       S41            S42            S43            S44\n");
fprintf(f, "weight  # of  weight  # of  weight  # of  weight  # of  weight\n");
fprintf(f, "range  axles distri. axles distri. axles distri. axles distri.\n");
for (i=0; i<MAP1; i++) {
        ppp1 = s41[i] / summ1;
        ppp2 = s42[i] / summ2;
        ppp3 = s43[i] / summ3;
        ppp4 = s44[i] / summ4;
        ffs41 += ppp1 * factor1[i];
        ffs42 += ppp2 * factor1[i];
        ffs43 += ppp3 * factor1[i];
        ffs44 += ppp4 * factor1[i];
        fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf%8ld%8.2lf%8ld%8.2lf\n", i*2 + 2,
        s41[i], ppp1 * 100.0, s42[i], ppp2 * 100.0,
        s43[i], ppp3 * 100.0, s44[i], ppp4 * 100.0);
}
fprintf(f, "total%7ld%16ld%16ld%16ld\n", sum1, sum2, sum3, sum4);

sum1 = get_sum(t42, MAP2, &summ1);
fprintf(f, "\n       T4\n");
fprintf(f, "weight  # of  weight\n");
fprintf(f, "range  axles distri.\n");
for (i=0; i<MAP2; i++) {
        ppp1 = t42[i] / summ1;
        fft42 += ppp1 * factor2[i];
        fprintf(f, "%4d%8ld%8.2lf\n", i*4 + 4, t42[i], ppp1 * 100.0);
}
fprintf(f, "total%7ld\n", sum1);
```

```
sum1 = get_sum(e41, MAP3, &summ1);
sum2 = get_sum(e42, MAP3, &summ2);
sum3 = get_sum(e43, MAP3, &summ3);
fprintf(f, "\n       E41          E42          E43\n");
fprintf(f, "weight  # of  weight  # of  weight  # of  weight\n");
fprintf(f, "range  axles distri. axles distri. axles distri.\n");
for (i=0; i<MAP3; i++) {
        ppp1 = e41[i] / summ1;
        ppp2 = e42[i] / summ2;
        ppp3 = e43[i] / summ3;
        fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf%8ld%8.2lf\n",
        i*2 + 13, e41[i], ppp1 * 100.0,
        e42[i], ppp2 * 100.0, e43[i], ppp3 * 100.0);
}
fprintf(f, "total%7ld%16ld%16ld\n", sum1, sum2, sum3);

// type 5
fprintf(f, "\nTYPE 5:\n");
sum1 = get_sum(t5, MAP2, &summ1);
fprintf(f, "       T5\n");
fprintf(f, "weight  # of  weight\n");
fprintf(f, "range  axles distri.\n");
for (i=0; i<MAP2; i++) {
        ppp1 = t5[i] / summ1;
        fft5 += ppp1 * factor2[i];
        fprintf(f, "%4d%8ld%8.2lf\n", i*4 + 4, t5[i], ppp1 * 100.0);
}
fprintf(f, "total%7ld\n", sum1);

sum1 = get_sum(e5, MAP3, &summ1);
fprintf(f, "\n       E5\n");
fprintf(f, "weight  # of  weight\n");
fprintf(f, "range  axles distri.\n");
for (i=0; i<MAP3; i++) {
        fprintf(f, "%4d%8ld%8.2lf\n", i*2 + 13, e5[i], e5[i]/summ1);
}
fprintf(f, "total%7ld\n", sum1);

// type 6
fprintf(f, "\nTYPE 6:\n");
sum1 = get_sum(s6, MAP1, &summ1);
fprintf(f, "       S6\n");
fprintf(f, "weight  # of  weight\n");
fprintf(f, "range  axles distri.\n");
for (i=0; i<MAP1; i++) {
```

```
        ppp1 = s6[i]/summ1;
        ffs6 += ppp1 * factor1[i];
        fprintf(f, "%4d%8ld%8.2lf\n", i*2 + 2, s6[i], ppp1 * 100.0);
    }
    fprintf(f, "total%7ld\n", sum1);

    sum1 = get_sum(t6, MAP2, &summ1);
    fprintf(f, "\n        T6\n");
    fprintf(f, "weight  # of   weight\n");
    fprintf(f, "range   axles  distri.\n");
    for (i=0; i<MAP2; i++) {
        ppp1 = t6[i]/summ1;
        fft6 += ppp1 * factor2[i];
        fprintf(f, "%4d%8ld%8.2lf\n", i*4 + 4, t6[i], ppp1 * 100.0);
    }
    fprintf(f, "total%7ld\n", sum1);

    sum1 = get_sum(e61, MAP3, &summ1);
    sum2 = get_sum(e62, MAP3, &summ2);
    fprintf(f, "\n        E61           E62\n");
    fprintf(f, "weight  # of   weight   # of   weight\n");
    fprintf(f, "range   axles  distri.  axles  distri.\n");
    for (i=0; i<MAP3; i++) {
        fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf\n", i*2 + 13,
        e61[i], e61[i]/summ1, e62[i], e62[i]/summ2);
    }
    fprintf(f, "total%7ld%16ld\n", sum1, sum2);

    sum1 = get_sum(r6, MAP4, &summ1);
    fprintf(f, "\n        R6\n");
    fprintf(f, "weight  # of   weight\n");
    fprintf(f, "range   axles  distri.\n");
    for (i=0; i<MAP4; i++) {
        ppp1 = r6[i]/summ1;
        ffr6 += ppp1 * factor3[i];
        fprintf(f, "%4d%8ld%8.2lf\n", i*4 + 4, r6[i], ppp1 * 100.0);
    }
    fprintf(f, "total%7ld\n", sum1);

// type 7
    fprintf(f, "\nTYPE 7:\n");
    sum1 = get_sum(s71, MAP1, &summ1);
    sum2 = get_sum(s72, MAP1, &summ2);
    fprintf(f, "        S71           S72\n");
    fprintf(f, "weight  # of   weight   # of   weight\n");
```

```
        fprintf(f, "range   axles  distri.  axles  distri.\n");
        for (i=0; i<MAP1; i++) {
                ppp1 = s71[i]/summ1;
                ppp2 = s72[i]/summ2;
                ffs71 += ppp1 * factor1[i];
                ffs72 += ppp2 * factor1[i];
                fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf\n", i*2 + 2,
                s71[i], ppp1 * 100.0, s72[i], ppp2 * 100.0);
        }
        fprintf(f, "total%7ld%16ld\n", sum1, sum2);

        sum1 = get_sum(e7, MAP3, &summ1);
        fprintf(f, "\n          E7\n");
        fprintf(f, "weight  # of   weight\n");
        fprintf(f, "range   axles  distri.\n");
        for (i=0; i<MAP3; i++) {
                fprintf(f, "%4d%8ld%8.2lf\n", i*2 + 13, e7[i], e7[i]/summ1);
        }
        fprintf(f, "total%7ld\n", sum1);

// type 8
        fprintf(f, "\nTYPE 8:\n");
        sum1 = get_sum(s81, MAP1, &summ1);
        sum2 = get_sum(s82, MAP1, &summ2);
        fprintf(f, "        S81            S82\n");
        fprintf(f, "weight  # of   weight   # of   weight\n");
        fprintf(f, "range   axles  distri.  axles  distri.\n");
        for (i=0; i<MAP1; i++) {
                ppp1 = s81[i] / summ1;
                ppp2 = s82[i] / summ2;
                ffs81 += ppp1 * factor1[i];
                ffs82 += ppp2 * factor1[i];
                fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf\n", i*2 + 2,
                s81[i], ppp1 * 100.0, s82[i], ppp2 * 100.0);
        }
        fprintf(f, "total%7ld%16ld\n", sum1, sum2);

        sum1 = get_sum(t81, MAP2, &summ1);
        sum2 = get_sum(t82, MAP2, &summ2);
        fprintf(f, "\n        T81            T82\n");
        fprintf(f, "weight  # of   weight   # of   weight\n");
        fprintf(f, "range   axles  distri.  axles  distri.\n");
        for (i=0; i<MAP2; i++) {
                ppp1 = t81[i]/summ1;
                ppp2 = t82[i]/summ2;
```

```
            fft81 += ppp1 * factor2[i];
            fft82 += ppp2 * factor2[i];
            fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf\n", i*4 + 4,
            t81[i], ppp1 * 100.0, t82[i], ppp2 * 100.0);
        }
        fprintf(f, "total%7ld%16ld\n", sum1, sum2);

        sum1 = get_sum(e81, MAP3, &summ1);
        sum2 = get_sum(e82, MAP3, &summ2);
        fprintf(f, "\n       E81           E82\n");
        fprintf(f, "weight  # of   weight   # of   weight\n");
        fprintf(f, "range   axles  distri.  axles  distri.\n");
        for (i=0; i<MAP3; i++) {
            fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf\n", i*2 + 13,
            e81[i], e81[i] / summ1, e82[i], e82[i] / summ2);
        }
        fprintf(f, "total%7ld%16ld\n", sum1, sum2);

// type 9
        fprintf(f, "\nTYPE 9:\n");
        sum1 = get_sum(s9, MAP1, &summ1);
        sum2 = get_sum(ds92, MAP1, &summ2);
        sum3 = get_sum(ds92, MAP1, &summ3);
        fprintf(f, "        S9          DS91          DS92\n");
        fprintf(f, "weight  # of   weight   # of   weight   # of   weight\n");
        fprintf(f, "range   axles  distri.  axles  distri.  axles  distri.\n");
        for (i=0; i<MAP1; i++) {
            ppp1 = s9[i] / summ1;
            ppp2 = ds91[i] / summ2;
            ppp3 = ds92[i] / summ3;
            ffs9 += ppp1 * factor1[i];
            ffds91 += ppp2 * factor1[i];
            ffds92 += ppp3 * factor1[i];
            fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf%8ld%8.2lf\n", i*2 + 2,
            s9[i], ppp1 * 100.0, ds91[i], ppp2 * 100.0, ds92[i], ppp3 * 100.0);
        }
        fprintf(f, "total%7ld%16ld%16ld\n", sum1, sum2, sum3);

        sum1 = get_sum(t91, MAP2, &summ1);
        sum2 = get_sum(t92, MAP2, &summ2);
        sum3 = get_sum(dt9, MAP2, &summ3);
        fprintf(f, "\n       T91           T92           DT9\n");
        fprintf(f, "weight  # of   weight   # of   weight   # of   weight\n");
        fprintf(f, "range   axles  distri.  axles  distri.  axles  distri.\n");
        for (i=0; i<MAP2; i++) {
```

```
            ppp1 = t91[i] / summ1;
            ppp2 = t92[i] / summ2;
            ppp3 = dt9[i] / summ3;
            fft91 += ppp1 * factor2[i];
            fft92 += ppp2 * factor2[i];
            ffdt9 += ppp3 * factor2[i];
            fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf%8ld%8.2lf\n", i*4 + 4,
            t91[i], ppp1 * 100.0, t92[i], ppp2 * 100.0, dt9[i], ppp3 * 100.0);
    }
    fprintf(f, "total%7ld%16ld%16ld\n", sum1, sum2, sum3);

    sum1 = get_sum(e91, MAP3, &summ1);
    sum2 = get_sum(e92, MAP3, &summ2);
    sum3 = get_sum(e93, MAP3, &summ3);
    fprintf(f, "\n       E91             E92             E93\n");
    fprintf(f, "weight  # of  weight  # of  weight  # of  weight\n");
    fprintf(f, "range   axles distri. axles distri. axles distri.\n");
    for (i=0; i<MAP3; i++) {
            fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf%8ld%8.2lf\n", i*2 + 13,
            e91[i], e91[i] / summ1, e92[i], e92[i] / summ2,
            e93[i], e93[i] / summ3);
    }
    fprintf(f, "total%7ld%16ld%16ld\n", sum1, sum2, sum3);

    sum1 = get_sum(r9, MAP4, &summ1);
    fprintf(f, "\n        R9\n");
    fprintf(f, "weight  # of  weight\n");
    fprintf(f, "range   axles distri.\n");
    for (i=0; i<MAP4; i++) {
            ppp1 = r9[i] / summ1;
            ffr9 += ppp1 * factor3[i];
            fprintf(f, "%4d%8ld%8.2lf\n", i*4 + 4, r9[i], ppp1 * 100.0);
    }
    fprintf(f, "total%7ld\n", sum1);

// type 10
    fprintf(f, "\nTYPE 10:\n");
    sum1 = get_sum(t10, MAP2, &summ1);
    fprintf(f, "        T10\n");
    fprintf(f, "weight  # of  weight\n");
    fprintf(f, "range   axles distri.\n");
    for (i=0; i<MAP2; i++) {
            ppp1 = t10[i] / summ1;
            fft10 += ppp1 * factor1[i];
            fprintf(f, "%4d%8ld%8.2lf\n", i*4 + 4, t10[i], ppp1 * 100.0);
```

```
	}
	fprintf(f, "total%7ld\n", sum1);

	sum1 = get_sum(e10, MAP3, &summ1);
	fprintf(f, "\n        E10\n");
	fprintf(f, "weight  # of   weight\n");
	fprintf(f, "range   axles  distri.\n");
	for (i=0; i<MAP3; i++) {
		fprintf(f, "%4d%8ld%8.2lf\n", i*2 + 13, e10[i], e10[i] / summ1);
	}
	fprintf(f, "total%7ld\n", sum1);

	sum1 = get_sum(r10, MAP4, &summ1);
	fprintf(f, "\n        R10\n");
	fprintf(f, "weight  # of   weight\n");
	fprintf(f, "range   axles  distri.\n");
	for (i=0; i<MAP4; i++) {
		ppp1 = r10[i] / summ1;
		ffr10 += ppp1 * factor3[i];
		fprintf(f, "%4d%8ld%8.2lf\n", i*4 + 4, r10[i], ppp1 * 100.0);
	}
	fprintf(f, "total%7ld\n", sum1);

// type 11
	fprintf(f, "\nTYPE 11:\n");
	sum1 = get_sum(s111, MAP1, &summ1);
	sum2 = get_sum(s112, MAP1, &summ2);
	sum3 = get_sum(s113, MAP1, &summ3);
	sum4 = get_sum(s114, MAP1, &summ4);
	fprintf(f, "        S111           S112           S113           S114\n");
	fprintf(f, "weight  # of   weight  # of   weight  # of   weight  # of   weight\n");
	fprintf(f, "range   axles  distri. axles  distri. axles  distri. axles  distri.\n");
	for (i=0; i<MAP1; i++) {
		ppp1 = s111[i] / summ1;
		ppp2 = s112[i] / summ2;
		ppp3 = s113[i] / summ3;
		ppp4 = s114[i] / summ4;
		ffs111 += ppp1 * factor1[i];
		ffs112 += ppp2 * factor1[i];
		ffs113 += ppp3 * factor1[i];
		ffs114 += ppp4 * factor1[i];
		fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf%8ld%8.2lf%8ld%8.2lf\n",
		i*2 + 2, s111[i], ppp1 * 100.0, s112[i], ppp2 * 100.0,
		s113[i], ppp3 * 100.0, s114[i], ppp4 * 100.0);
	}
```

```
	fprintf(f, "total%7ld%16ld%16ld%16ld\n", sum1, sum2, sum3, sum4);

	sum1 = get_sum(e11, MAP3, &summ1);
	fprintf(f, "\n      E11\n");
	fprintf(f, "weight  # of  weight\n");
	fprintf(f, "range  axles  distri.\n");
	for (i=0; i<MAP3; i++) {
		fprintf(f, "%4d%8ld%8.2lf\n", i*2 + 13, e11[i], e11[i]/summ1);
	}
	fprintf(f, "total%7ld\n", sum1);

// type 12
	fprintf(f, "\nTYPE 12:\n");
	sum1 = get_sum(s1211, MAP1, &summ1);
	sum2 = get_sum(s1212, MAP1, &summ2);
	sum3 = get_sum(s1213, MAP1, &summ3);
	sum4 = get_sum(s1221, MAP1, &summ4);
	sum5 = get_sum(s1222, MAP1, &summ5);
	sum6 = get_sum(s1223, MAP1, &summ6);
	fprintf(f, "     S1211          S1212          S1213\n");
	fprintf(f, "weight  # of  weight  # of  weight  # of  weight\n");
	fprintf(f, "range  axles  distri.  axles  distri.  axles  distri.\n");
	for (i=0; i<MAP1; i++) {
		ppp1 = s1211[i] / summ1;
		ppp2 = s1212[i] / summ2;
		ppp3 = s1213[i] / summ3;
		ffs1211 += ppp1 * factor1[i];
		ffs1212 += ppp2 * factor1[i];
		ffs1213 += ppp3 * factor1[i];
		fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf%8ld%8.2lf\n",
		i*2 + 2, s1211[i], ppp1 * 100.0, s1212[i], ppp2 * 100.0,
		s1213[i], ppp3 * 100.0);
	}
	fprintf(f, "total%7ld%16ld%16ld\n", sum1, sum2, sum3);

	fprintf(f, "\n     S1221          S1222          S1223\n");
	fprintf(f, "weight  # of  weight  # of  weight  # of  weight\n");
	fprintf(f, "range  axles  distri.  axles  distri.  axles  distri.\n");
	for (i=0; i<MAP1; i++) {
		ppp1 = s1221[i] / summ4;
		ppp2 = s1222[i] / summ5;
		ppp3 = s1223[i] / summ6;
		ffs1221 += ppp1 * factor1[i];
		ffs1222 += ppp2 * factor1[i];
		ffs1223 += ppp3 * factor1[i];
```

```
            fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf%8ld%8.2lf\n",
            i*2 + 2, s1221[i], ppp1 * 100.0,
            s1222[i], ppp2 * 100.0, s1223[i], ppp3 * 100.0);
        }
        fprintf(f, "total%7ld%16ld%16ld\n", sum4, sum5, sum6);

        sum1 = get_sum(t121, MAP2, &summ1);
        sum2 = get_sum(t122, MAP2, &summ2);
        fprintf(f, "\n       T121           T122\n");
        fprintf(f, "weight  # of  weight  # of  weight\n");
        fprintf(f, "range   axles  distri.  axles  distri.\n");
        for (i=0; i<MAP2; i++) {
            ppp1 = t121[i]/summ1;
            ppp2 = t122[i]/summ2;
            fft121 += ppp1 * factor2[i];
            fft122 += ppp2 * factor2[i];
            fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf\n", i*4 + 4,
            t121[i], ppp1 * 100.0, t122[i], ppp2 * 100.0);
        }
        fprintf(f, "total%7ld%16ld\n", sum1, sum2);

        sum1 = get_sum(e121, MAP3, &summ1);
        sum2 = get_sum(e122, MAP3, &summ2);
        fprintf(f, "\n       E121           E122\n");
        fprintf(f, "weight  # of  weight  # of  weight\n");
        fprintf(f, "range   axles  distri.  axles  distri.\n");
        for (i=0; i<MAP3; i++) {
            fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf\n", i*2 + 13,
            e121[i], e121[i]/summ1, e122[i], e122[i]/summ2);
        }
        fprintf(f, "total%7ld%16ld\n", sum1, sum2);

// type 13
        fprintf(f, "\nTYPE 13:\n");
        sum1 = get_sum(s131, MAP1, &summ1);
        sum2 = get_sum(s132, MAP1, &summ2);
        fprintf(f, "       S131           S132\n");
        fprintf(f, "weight  # of  weight  # of  weight\n");
        fprintf(f, "range   axles  distri.  axles  distri.\n");
        for (i=0; i<MAP1; i++) {
            ppp1 = s131[i] / summ1;
            ppp2 = s132[i] / summ2;
            ffs131 += ppp1 * factor1[i];
            ffs132 += ppp2 * factor1[i];
            fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf\n", i*2 + 2,
```

```
        s131[i], ppp1 * 100.0, s132[i], ppp2 * 100.0);
}
fprintf(f, "total%7ld%16ld\n", sum1, sum2);

sum1 = get_sum(t131, MAP2, &summ1);
sum2 = get_sum(t132, MAP2, &summ2);
fprintf(f, "\n      T131           T132\n");
fprintf(f, "weight  # of  weight  # of  weight\n");
fprintf(f, "range   axles  distri.  axles  distri.\n");
for (i=0; i<MAP2; i++) {
        ppp1 = t131[i]/summ1;
        ppp2 = t132[i]/summ2;
        fft131 += ppp1 * factor2[i];
        fft132 += ppp2 * factor2[i];
        fprintf(f, "%4d%8ld%8.2lf%8ld%8.2lf\n", i*4 + 4,
        t131[i], ppp1 * 100.0, t132[i], ppp2 * 100.0);
}
fprintf(f, "total%7ld%16ld\n", sum1, sum2);

sum1 = get_sum(e13, MAP3, &summ1);
fprintf(f, "\n       E13\n");
fprintf(f, "weight  # of  weight\n");
fprintf(f, "range   axles  distri.\n");
for (i=0; i<MAP3; i++) {
        fprintf(f, "%4d%8ld%8.2lf\n", i*2 + 13, e13[i], e13[i]/summ1);
}
fprintf(f, "total%7ld\n", sum1);

/* ouyput ff... */
fprintf(f, "\n--------------------------------------------\n");
fprintf(f, "\nType 4\n");
fprintf(f, " 2D : %lf\n", ffs41);
fprintf(f, " 2D-1 : %lf\n", ffs42 + ffs43);
fprintf(f, " 2D-2 : %lf\n", ffs44 + fft42);

fprintf(f, "\nType 5\n");
fprintf(f, " 3A : %lf\n", fft5);

fprintf(f, "\nType 6\n");
fprintf(f, " 4A : %lf\n", ffr6);
fprintf(f, " Rig : %lf\n", ffs6 + fft6);

fprintf(f, "\nType 7\n");
fprintf(f, " 2S1 : %lf\n", ffs71 + ffs72);
```

```
        fprintf(f, "\nType 8\n");
        fprintf(f, " 2S2 : %lf\n", ffs81 + fft81);
        fprintf(f, " 3S1 : %lf\n", fft82 + ffs82);

        fprintf(f, "\nType 9\n");
        fprintf(f, " 2S3 : %lf\n", ffs9 + ffr9);
        fprintf(f, " 3S3 : %lf\n", fft91 + fft92);
        fprintf(f, " 3S2 : %lf\n", ffdt9 + ffds91 + ffds92);

        fprintf(f, "\nType 10\n");
        fprintf(f, " 3S3 : %lf\n", fft10 + ffr10);

        fprintf(f, "\nType 11\n");
        fprintf(f, " 2S1-2 : %lf\n", ffs111 + ffs112 + ffs113 + ffs114);

        fprintf(f, "\nType 12\n");
        fprintf(f, " 2S2-2 : %lf\n", ffs1211 + fft121 + ffs1212 + ffs1213);
        fprintf(f, " 3S1-2 : %lf\n", ffs1221 + fft122 + ffs1222 + ffs1223);

        fprintf(f, "\nType 13\n");
        fprintf(f, " 3S2-2 : %lf\n", fft131 + fft132 + ffs131 + ffs132);

        fclose(f);
}

static long get_sum(long *tt, int num, double *ss)
{
        int  i;
        long sum = 0;

        for (i=0; i<num; i++) {
                sum += tt[i];
        }
        if (sum == 0)
                *ss = 1.0;
        else
                *ss = (double)sum;
        return sum;
}

static double get_beta(double L1, double L2)
{
        double beta;

        if (pave == 1) {
```

```
		beta = 0.40 + (0.081 * pow(L1 + L2, 3.23)) /
		pow(sn + 1.0, 5.19) / pow(L2, 3.23);
	} else {
		beta = 1.0 + (3.63 * pow(L1 + L2, 5.2)) /
		(pow(sn + 1.0, 8.46) * pow(L2, 3.52));
	}
	return beta;
}

static double get_wt(double L1, double L2)
{
	double wt;
	double beta = get_beta(L1, L2);

	if (pave == 1) {
		wt = 5.93 + 9.36 * log10(sn + 1.0) - 4.79 * log10(L1 + L2) +
			4.331 * log10(L2) + gt / beta;
	} else {
		wt = 5.85 + 7.35 * log10(sn + 1.0) - 4.62 * log10(L1 + L2) +
			3.28 * log10(L2) + gt / beta;
	}
	wt = pow(10, wt);
	return wt;
}
static double get_factor(double L1, double L2)
{
	double beta = get_beta(L1, L2);
	double t1;
	double t2;
	double wt = get_wt(L1, L2);

	if (pave == 1) {
		t1 = pow(L1 + L2, 4.79) / pow(19.0, 4.79);
		t2 = pow(10.0, gt / beta18) / pow(10.0, gt / beta) / pow(L2, 4.331);
	} else {
		t1 = pow(L1 + L2, 4.62) / pow(19.0, 4.62);
		t2 = pow(10.0, gt / beta18) / pow(10.0, gt / beta) / pow(L2, 3.28);
	}
/*	wt = t1 * t2; */
	t1 *= t2;
/*	wt = wt18 / wt;
	wt *= t1; */
	return t1;
}
```

A.2 ESAL PROGRAM

COMPUTER PROGRAM FOR FORECASTING FUTURE TRAFFIC LOADS

```
#INCLUDE <STDIO.H>
#INCLUDE <STDLIB.H>
#INCLUDE <STRING.H>
#INCLUDE <DOS.H>
#INCLUDE <MATH.H>

#DEFINE MIN_TYPE  4
#DEFINE MAX_TYPE  13
STATIC DOUBLE FFF[]={0.0, 0.0, 0.0, 0.0, 0.031, 0.178, 0.297, 0.277,
                        0.261, 0.567, 0.851, 1.521, 0.615, 0.745};

MAIN()
{
        INT             I, N, GR;
        LONG CURR_TRAF[15];
        DOUBLE          G[15];
        DOUBLE          FACTOR[15];
        DOUBLE          FF;
        LONG    DESIGN_TRAF, DESIGN_ESAL;
        LONG    SUM1, SUM2, SUM3;
        CHAR FILENAME[100];
        FILE    *F;

        PRINTF("ANALYSIS PERIOD (YEAR): ");
        SCANF("%D", &N);
        printf("1: simple growth rate, 2: compound growth rate. :");
        SCANF("%D", &GR);

        FOR (I = MIN_TYPE; I <= MAX_TYPE; I++) {
                PRINTF("CURRENT TRAFFIC PER DAY OF TYPE %D: ", I);
                SCANF("%LD", &CURR_TRAF[I]);
                PRINTF("ANNUAL GROWTH RATE OF TYPE %D (E.G. 0.05): ", I);
                SCANF("%LF", &G[I]);
                PRINTF("E.S.A.L. FACTOR PER VEHICLE OF TYPE %D (0 FOR
DEFAULT VALUE %.3LF): ", I, FFF[I]);
                SCANF("%LF", &FACTOR[I]);
                IF (FACTOR[I] == 0.0)
                        FACTOR[I] = FFF[I];
        }
        PRINTF("OUTPUT FILENAME: ");
        SCANF("%S", FILENAME);
        IF ((F = FOPEN(FILENAME, "WT")) == NULL) {
                PRINTF("CANNOT OPEN FILE %S\N", FILENAME);
                RETURN 0;
        }
```

```
DO {
        PRINTF("\N\N");
        PRINTF("TRUCK  CURRENT  GROWTH  E.S.A.L.\N");
        PRINTF(" TYPE   TRAFFIC  RATE    FACTOR\N");
        PRINTF("------------------------------------\N");
        FOR (I = MIN_TYPE; I <= MAX_TYPE; I++) {
                PRINTF("%4D%9LD%10.2LF%12.4LF\N",
                I, CURR_TRAF[I], G[I], FACTOR[I]);
        }
        PRINTF("------------------------------------\N\N");
        PRINTF("ENTER TRUCK TYPE TO BE MODIFIED (O FOR EXIT): ");
        SCANF("%D", &I);
        IF (I < MIN_TYPE || I > MAX_TYPE)
                BREAK;
        PRINTF("CURRENT TRAFFIC PER DAY OF TYPE %D: ", I);
        SCANF("%LD", &CURR_TRAF[I]);
        PRINTF("ANNUAL GROWTH RATE OF TYPE %D: ", I);
        SCANF("%LF", &G[I]);;
        PRINTF("E.S.A.L. FACTOR PER VEHICLE OF TYPE %D: ", I);
        SCANF("%LF", &FACTOR[I]);
        IF (FACTOR[I] == 0.0)
                FACTOR[I] = FFF[I];
} WHILE (I != 0);

SUM1 = SUM2 = SUM3 = 0;
FPRINTF(F, "TRUCK  CURRENT  GROWTH   DESIGN   E.S.A.L.
DESIGN\N");
FPRINTF(F, "TYPES  TRAFFIC   RATE    TRAFFIC   FACTOR
E.S.A.L.\N");
FPRINTF(F, "     (AADT)        (CUMULATION)  (/VEH.)\N");
FPRINTF(F, "------------------------------------------------------------\N");
FOR (I = MIN_TYPE; I <= MAX_TYPE; I++) {
        IF (G[I] == 0.0) {
                FF = (DOUBLE)N;
        } ELSE IF (GR == 1) {
                FF = (DOUBLE)N + (DOUBLE)N*(DOUBLE)(N-1)*G[I] / 2.0;
        } ELSE {
                FF = (POW(1.0 + G[I], (DOUBLE)N) - 1) / G[I];
        }
        DESIGN_TRAF = (LONG)(CURR_TRAF[I] * FF) * 365;
        DESIGN_ESAL = (LONG)(DESIGN_TRAF * FACTOR[I]);
        SUM1 += CURR_TRAF[I];
        SUM2 += DESIGN_TRAF;
        SUM3 += DESIGN_ESAL;
```

```
        FPRINTF(F, "%4D%9LD%7D%%%15LD%12.4LF%12LD\N",
                I, CURR_TRAF[I], (INT)(G[I] * 100.0), DESIGN_TRAF,
                FACTOR[I], DESIGN_ESAL);
    }
    FPRINTF(F, "------------------------------------------------------------\N");
    FPRINTF(F, "TOTAL%8LD%23LD%24LD\N", SUM1, SUM2, SUM3);
    FCLOSE(F);
    RETURN 0;
}
```

APPENDIX B:

SOUTH SITE TRAFFIC COUNT FOR 1995

(N/A = Information not available)

TRAFFIC VOLUME OF JANUARY 1995, SOUTH SITE

DATE	WEEK	TYPE I	TYPE 2	TYPE 3	TYPE 4	TYPE 5	TYPE 6	TYPE 7	TYPE 8	TYPE 9	TYPE 10	TYPE 11	TYPE 12	TYPE 13	TYPE 14	TOTAL
1-JAN	SUN	3981	2414	12	487	9	0	23	65	289	0	23	3	0	14	7320
2-JAN	MON	4252	2523	14	586	55	0	38	83	807	7	28	3	0	22	8418
3-JAN	TUE	2697	1886	38	440	48	0	32	98	1236	11	49	6	1	26	6568
4-JAN	WED	2458	1718	30	462	55	4	27	89	1296	12	52	10	0	16	6229
5-JAN	THU	2355	1609	25	422	61	0	23	104	1409	14	64	13	1	25	6125
6-JAN	FRI	2773	1826	35	498	44	2	31	99	1141	16	46	7	0	25	6543
7-JAN	SAT	3034	1873	14	448	40	0	24	53	527	16	48	7	1	13	6098
8-JAN	SUN	3248	1801	16	424	28	0	18	54	708	7	20	3	1	28	6356
9-JAN	MON	2362	1677	32	455	59	0	29	98	1422	19	46	7	1	27	6234
10-JAN	TUE	2333	1611	26	424	61	1	32	94	1395	14	54	8	1	26	6080
11-JAN	WED	2413	1615	33	485	66	0	28	99	1425	31	55	7	3	33	6293
12-JAN	THU	2516	1721	37	513	64	1	47	126	1467	31	70	8	1	35	6637
13-JAN	FRI	3282	2076	32	539	59	1	31	79	1169	22	60	8	0	18	7376
14-JAN	SAT	3185	2094	15	487	35	0	27	86	567	7	52	10	1	19	6585
15-JAN	SUN	3321	2038	12	465	25	0	26	63	692	11	34	1	0	20	6708
16-JAN	MON	2984	1864	20	504	50	2	36	102	1389	17	46	11	0	25	7050
17-JAN	TUE	2276	1515	32	432	52	1	28	88	1298	21	54	7	0	19	5823
18-JAN	WED	1984	1443	28	394	54	1	34	96	1383	10	70	11	0	23	5531
19-JAN	THU	2412	1624	25	420	53	1	35	100	1493	18	71	13	0	20	6285
20-JAN	FRI	3264	2003	30	542	55	0	27	94	1205	13	61	12	1	32	7339
21-JAN	SAT	2997	1841	16	427	48	0	21	50	542	10	57	9	0	18	6036
22-JAN	SUN	3459	2099	20	500	29	0	27	58	718	5	38	2	0	15	6970
23-JAN	MON	2217	1565	33	428	68	1	27	79	1326	17	38	9	1	24	5833
24-JAN	TUE	2092	1489	23	430	65	1	27	77	1299	19	55	10	1	18	5606
25-JAN	WED	2166	1578	33	443	63	1	31	84	1398	13	55	10	1	22	5898
26-JAN	THU	2250	1589	26	398	56	1	21	103	1433	10	68	19	1	34	6009
27-JAN	FRI	3176	2017	35	642	51	0	34	98	1137	11	57	12	1	36	7307
28-JAN	SAT	3007	1880	18	486	37	0	27	79	513	5	57	10	1	14	6134
29-JAN	SUN	3113	1957	12	528	16	0	32	67	706	8	33	4	0	26	6502
30-JAN	MON	2327	1720	33	496	58	0	30	78	1335	16	58	10	1	20	6172
31-JAN	TUE	2150	1582	33	426	80	2	34	98	1271	12	65	10	0	30	5793

TRAFFIC VOLUME OF FEBUARY 1995, SOUTH SITE

DATE	WEEK	TYPE I	TYPE 2	TYPE 3	TYPE 4	TYPE 5	TYPE 6	TYPE 7	TYPE 8	TYPE 9	TYPE 10	TYPE 11	TYPE 12	TYPE 13	TYPE 14	TOTAL
1-FEB	WED	2219	1598	38	452	72	1	29	96	1442	20	65	16	1	23	6072
2-FEB	THU	2412	1714	31	444	99	4	32	97	1436	19	78	9	0	27	6402
3-FEB	FRI	3440	2149	34	644	94	2	32	96	1192	12	70	10	0	36	7811
4-FEB	SAT	3323	2056	15	485	47	0	37	96	505	11	51	6	0	17	6649
5-FEB	SUN	4086	2429	25	689	33	0	50	109	734	6	37	6	2	28	8234
6-FEB	MON	2376	1762	30	507	76	0	35	89	1348	13	51	5	1	25	6318
7-FEB	TUE	2089	1610	34	477	77	0	26	89	1301	24	51	7	1	27	5813
8-FEB	WED	2260	1560	34	456	59	0	31	86	1459	17	61	13	1	21	6058
9-FEB	THU	2402	1730	29	505	81	0	35	104	1524	21	69	8	0	24	6532
10-FEB	FRI	3642	2178	37	674	55	2	30	109	1289	22	57	13	0	29	8137
11-FEB	SAT	3064	1904	20	491	37	1	34	56	569	6	60	5	0	14	6261
12-FEB	SUN	3679	2066	13	524	18	0	30	76	720	6	40	5	1	23	7201
13-FEB	MON	2261	1608	21	441	70	0	22	88	1314	22	59	7	1	21	5935
14-FEB	TUE	2113	1509	39	453	65	0	16	85	1286	16	52	6	0	32	5671
15-FEB	WED	2205	1569	26	451	70	0	31	101	1356	20	64	14	0	24	5931
16-FEB	THU	2557	1776	36	534	80	0	36	100	1453	15	70	13	0	28	6698
17-FEB	FRI	3768	2225	45	712	91	0	38	92	1143	14	63	12	0	48	8251
18-FEB	SAT	3201	1991	23	479	54	0	24	72	573	6	46	8	0	20	6497
19-FEB	SUN	3848	2294	13	600	20	0	40	80	694	6	38	5	1	35	7674
20-FEB	MON	3153	2033	34	536	67	0	51	111	1396	11	54	11	0	29	7486
21-FEB	TUE	2536	1806	43	451	82	0	24	99	1373	12	64	11	0	32	6533
22-FEB	WED	2278	1576	26	491	73	1	30	107	1563	28	57	12	0	21	6263
23-FEB	THU	2475	1722	46	513	85	1	23	125	1576	28	64	11	0	20	6689
24-FEB	FRI	3800	2243	49	752	70	0	37	105	1288	17	67	20	0	42	8490
25-FEB	SAT	3198	1986	28	532	49	0	39	76	626	9	53	7	0	21	6624
26-FEB	SUN	3716	2336	27	640	32	0	48	76	800	4	33	6	1	32	7751
27-FEB	MON	2359	1674	35	478	52	0	40	93	1450	18	69	7	2	35	6312
28-FEB	TUE	2153	1636	38	532	60	0	26	98	1300	12	63	14	0	31	5963

TRAFFIC VOLUME OF MARCH 1995, SOUTH SITE

DATE	WEEK	TYPE I	TYPE 2	TYPE 3	TYPE 4	TYPE 5	TYPE 6	TYPE 7	TYPE 8	TYPE 9	TYPE 10	TYPE 11	TYPE 12	TYPE 13	TYPE 14	TOTAL
1-MAR	WED	2248	1617	40	486	87	0	33	102	1409	32	68	13	0	29	6164
2-MAR	THU	2494	1767	42	501	87	1	27	76	1531	28	78	14	0	30	6676
3-MAR	FRI	3652	2089	39	559	51	2	29	94	1204	22	68	10	0	25	7844
4-MAR	SAT	3250	1955	23	471	22	0	24	75	575	8	48	6	0	12	6469
5-MAR	SUN	3701	2181	14	521	19	0	47	69	766	10	37	5	0	22	7392
6-MAR	MON	2459	1700	36	458	62	0	36	72	1490	17	51	8	0	17	6406
7-MAR	TUE	1993	1540	34	390	34	0	23	85	1288	14	66	13	0	16	5496
8-MAR	WED	2314	1613	32	447	88	0	32	86	1443	19	62	17	0	16	6169
9-MAR	THU	2665	1778	47	500	92	0	34	95	1591	21	75	13	2	35	6948
10-MAR	FRI	4486	2368	45	684	78	0	29	117	1300	21	63	9	1	37	9238
11-MAR	SAT	4238	2400	36	538	34	0	41	71	573	13	48	5	1	20	8018
12-MAR	SUN	4439	2526	17	590	22	1	79	86	691	7	37	8	2	30	8535
13-MAR	MON	2528	1621	34	414	5	2	46	72	1345	10	65	11	0	28	6230
14-MAR	TUE	2660	1740	26	420	47	1	35	101	1237	20	66	8	1	29	6391
15-MAR	WED	3082	1804	24	447	52	0	43	109	1467	21	57	12	0	20	7139
16-MAR	THU	3293	2007	33	520	79	0	29	90	1472	20	73	13	0	25	7654
17-MAR	FRI	4057	2338	22	557	73	0	51	105	1171	22	66	11	0	28	8501
18-MAR	SAT	4249	2465	25	542	46	0	39	79	552	14	51	12	0	28	8102
19-MAR	SUN	5383	3006	14	666	28	0	95	118	739	12	44	3	0	40	10148
20-MAR	MON	2756	1827	34	488	58	0	36	81	1484	19	48	8	0	25	6864
21-MAR	TUE	2453	1670	42	446	82	0	30	99	1356	21	53	8	1	28	6289
22-MAR	WED	2564	1767	35	453	62	1	25	114	1419	19	64	14	0	22	6559
23-MAR	THU	2729	1920	43	458	72	1	44	117	1558	34	78	10	1	27	7092
24-MAR	FRI	3823	2262	30	545	56	1	45	125	1283	15	67	12	0	30	8294
25-MAR	SAT	3775	2311	26	517	25	0	42	86	538	7	46	9	0	24	7406
26-MAR	SUN	4683	2621	20	630	21	0	91	94	743	9	37	0	0	43	8992
27-MAR	MON	2489	1946	40	488	46	0	41	106	1405	21	66	5	0	44	6597
28-MAR	TUE	2295	1605	55	454	85	1	27	127	1286	16	56	14	3	51	6075
29-MAR	WED	2296	1555	39	439	52	1	32	89	1449	24	76	11	0	22	6085
30-MAR	THU	2624	1770	34	469	52	2	29	106	1471	23	81	19	0	30	6710
31-MAR	FRI	3551	2096	43	574	56	2	35	100	1170	19	71	8	1	30	7756

TRAFFIC VOLUME OF APRIL 1995, SOUTH SITE

DATE	WEEK	TYPE I	TYPE 2	TYPE 3	TYPE 4	TYPE 5	TYPE 6	TYPE 7	TYPE 8	TYPE 9	TYPE 10	TYPE 11	TYPE 12	TYPE 13	TYPE 14	TOTAL
1-APR	SAT	3380	1997	25	429	38	0	38	72	564	10	57	7	0	11	6628
2-APR	SUN	4335	2509	26	615	27	2	63	85	792	8	37	1	0	38	8538
3-APR	MON	2542	1801	40	481	82	1	27	91	1415	24	64	6	1	22	6587
4-APR	TUE	2172	1555	38	427	41	1	23	83	1292	9	68	9	0	19	5737
5-APR	WED	2293	1607	30	429	56	0	29	85	1349	17	71	13	0	30	6009
6-APR	THU	2681	1825	37	471	65	2	31	107	1453	25	76	8	0	15	6796
7-APR	FRI	3649	2112	53	597	53	0	30	91	1141	17	63	9	2	37	7854
8-APR	SAT	3576	1993	26	432	52	0	30	61	576	10	48	8	0	18	6830
9-APR	SUN	4294	2633	19	585	16	0	66	107	715	7	39	2	0	35	8518
10-APR	MON	2283	1690	39	430	66	0	31	96	1323	12	64	7	0	19	6060
11-APR	TUE	2313	1671	43	422	61	0	31	77	1307	15	50	8	0	19	6017
12-APR	WED	2481	1682	33	503	62	1	42	95	1480	37	68	11	2	36	6533
13-APR	TUE	4065	2356	41	740	79	0	37	113	1503	23	67	16	0	60	9100
14-APR	FRI	3757	2081	105	1035	109	1	51	123	915	17	69	11	0	77	8351
15-APR	SAT	3879	2247	20	500	19	0	40	59	476	4	42	8	0	14	7308
16-APR	SUN	6841	3556	8	1017	20	0	73	124	512	2	43	6	0	52	12254
17-APR	MON	4172	2570	35	704	63	1	48	117	1304	22	49	6	0	33	9124
18-APR	TUE	2435	1700	29	474	70	1	27	60	1310	24	34	10	0	22	6196
19-APR	WED	2466	1613	27	473	76	5	28	107	1472	17	67	9	0	30	6390
20-APR	TUE	2595	1671	41	534	61	1	38	102	1582	11	63	11	0	22	6732
21-APR	FRI	3417	2177	33	601	45	2	33	98	1297	19	75	9	2	24	7832
22-APR	SAT	3212	1891	18	479	34	0	32	70	587	7	54	6	2	25	6417
23-APR	SUN	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
24-APR	MON	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
25-APR	TUE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
26-APR	WED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
27-APR	TUE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
28-APR	FRI	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
29-APR	SAT	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
30-APR	SUN	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

TRAFFIC VOLUME OF MAY 1995, SOUTH SITE

DATE	WEEK	TYPE 1	TYPE 2	TYPE 3	TYPE 4	TYPE 5	TYPE 6	TYPE 7	TYPE 8	TYPE 9	TYPE 10	TYPE 11	TYPE 12	TYPE 13	TYPE 14	TOTAL
1-MAY	MON	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2-MAY	TUE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
3-MAY	WED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
4-MAY	THU	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
5-MAY	FRI	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
6-MAY	SAT	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
7-MAY	SUN	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
8-MAY	MON	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
9-MAY	TUE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10-MAY	WED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
11-MAY	THU	3058	1942	55	496	87	0	50	107	1635	28	63	4	2	22	7549
12-MAYFRI	4135	2424	42	672	83	0	53	102	1315	20	72	11	1	27	8957	
13-MAYSAT	4528	2506	32	612	49	1	36	102	613	9	51	7	1	27	8574	
14-MAY	SUN	6165	3100	13	795	19	1	42	111	638	11	38	3	2	32	10970
15-MAY	MON	3101	1940	31	515	92	2	37	104	1427	10	51	6	0	32	7348
16-MAY	TUE	2528	1717	30	473	121	1	39	124	1398	16	58	6	1	51	6563
17-MAY	WED	2513	1639	40	449	127	0	43	105	1486	25	59	10	0	49	6544
18-MAY	THU	2604	1833	42	505	90	0	39	122	1504	32	73	14	0	35	6893
19-MAYFRI	3489	2130	30	591	89	2	38	106	1368	19	70	10	0	32	7974	
20-MAYSAT	3501	2127	36	473	50	1	32	70	597	19	48	7	0	21	6982	
21-MAY	SUN	4315	2489	25	579	19	0	67	112	735	12	37	3	0	36	8429
22-MAY	MON	2790	1777	40	472	88	0	37	107	1517	21	52	4	0	12	6917
23-MAY	TUE	2487	1647	41	437	88	0	28	85	1412	13	58	10	0	17	6323
24-MAY	WED	2645	1701	25	463	99	1	28	101	1535	19	68	11	2	31	6729
25-MAY	THU	3134	1955	36	524	100	2	28	116	1587	32	67	8	2	31	7622
26-MAYFRI	4543	2626	32	661	116	1	42	124	1333	17	61	8	2	49	9615	
27-MAYSAT	4518	2490	33	553	51	0	25	96	592	12	40	6	0	21	8427	
28-MAY	SUN	4752	2544	20	528	16	3	54	70	437	4	33	1	0	15	8477
29-MAY	MON	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
30-MAY	TUE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
31-MAY	WED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

TRAFFIC VOLUME OF JUNE 1995, SOUTH SITE

DATE	WEEK	TYPE 1	TYPE 2	TYPE 3	TYPE 4	TYPE 5	TYPE 6	TYPE 7	TYPE 8	TYPE 9	TYPE 10	TYPE 11	TYPE 12	TYPE 13	TYPE 14	TOTAL
1-JUN	THU	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2-JUN	FRI	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
3-JUN	SAT	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
4-JUN	SUN	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
5-JUN	MON	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
6-JUN	TUE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
7-JUN	WED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
8-JUN	THU	2640	1785	28	410	89	1	40	93	1503	19	58	11	1	37	6621
9-JUN	FRI	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10-JUN	SAT	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
11-JUN	SUN	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
12-JUN	MON	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
13-JUN	TUE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
14-JUN	WED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
15-JUN	THU	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
16-JUN	FRI	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
17-JUN	SAT	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
18-JUN	SUN	5761	3222	29	717	19	1	59	131	716	7	47	3	1	47	10759
19-JUN	MON	3259	2044	44	521	103	0	43	120	1436	25	51	5	1	39	7691
20-JUN	TUE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
21-JUN	WED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
22-JUN	THU	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
23-JUN	FRI	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
24-JUN	SAT	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
25-JUN	SUN	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
26-JUN	MON	2046	1023	0	585	146	0	0	0	1242	0	0	0	0	0	5850
27-JUN	TUE	2759	1873	33	454	98	1	42	112	1396	32	61	13	2	42	6918
28-JUN	WED	2908	1947	22	514	136	0	55	135	1573	23	72	8	1	28	7422
29-JUN	THU	2990	1907	29	579	97	0	37	135	1557	30	63	10	0	45	7479
30-JUN	FRI	4511	2637	34	719	83	1	48	121	1314	21	64	12	1	29	9595

TRAFFIC VOLUME OF JULY 1995, SOUTH SITE

DATE	WEEK	TYPE 1	TYPE 2	TYPE 3	TYPE 4	TYPE 5	TYPE 6	TYPE 7	TYPE 8	TYPE 9	TYPE 10	TYPE 11	TYPE 12	TYPE 13	TYPE 14	TOTAL
1-JUL	SAT	5167	2780	33	647	31	0	31	117	627	14	54	7	0	42	9550
2-JUL	SUN	5547	2968	23	634	16	0	61	118	712	7	41	1	0	28	10156
3-JUL	MON	4341	2387	31	549	83	0	49	123	941	13	41	5	0	37	8600
4-JUL	TUE	4251	2348	19	585	36	0	65	122	574	9	36	8	0	26	8079
5-JUL	WED	3705	2211	36	669	97	4	45	98	1332	20	53	7	2	34	8313
6-JUL	THU	3486	2220	36	535	106	1	36	130	1428	17	52	10	2	35	8094
7-JUL	FRI	4419	2533	37	602	103	0	53	153	1282	23	50	11	1	42	9308
8-JUL	SAT	4320	2540	34	542	63	0	40	118	597	8	47	7	0	19	8335
9-JUL	SUN	5357	2966	22	558	24	0	62	115	760	10	35	9	0	40	9958
10-JUL	MON	3198	2188	36	500	101	0	43	115	1422	19	50	7	0	26	7705
11-JUL	TUE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
12-JUL	WED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
13-JUL	THU	3095	1974	32	509	101	0	37	128	1537	30	69	14	1	26	7553
14-JUL	FRI	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
15-JUL	SAT	4200	2477	26	562	34	0	37	102	598	12	45	4	1	20	8118
16-JUL	SUN	5229	2987	22	784	19	1	64	122	765	11	39	3	0	38	10084
17-JUL	MON	3137	2015	30	517	112	1	45	133	1393	20	50	7	1	22	7483
18-JUL	TUE	2731	1706	116	717	254	3	93	155	1415	41	46	6	2	124	7057
19-JUL	WED	2664	561	476	1427	587	36	382	275	1544	168	19	16	15	688	7476
20-JUL	THU	2813	565	513	1463	561	33	426	265	1557	182	27	10	9	784	7805
21-JUL	FRI	3407	690	569	1719	462	27	405	259	1265	120	31	13	16	926	8765
22-JUL	SAT	2980	670	461	1576	195	10	352	206	634	48	20	9	6	764	7371
23-JUL	SUN	3560	774	515	1996	290	9	394	213	748	81	31	6	12	903	8870
24-JUL	MON	2758	569	487	1512	517	29	395	252	1366	134	23	16	7	741	7598
25-JUL	TUE	2779	571	476	1298	550	41	404	273	1378	142	26	13	11	743	7430
26-JUL	WED	2723	557	501	1288	516	40	382	246	1546	162	20	15	4	815	7395
27-JUL	THU	3032	596	509	1469	610	46	425	268	1659	175	27	17	11	786	8119
28-JUL	FRI	3792	747	593	1794	499	42	480	320	1315	141	23	21	15	849	9450
29-JUL	SAT	3276	792	477	1921	196	12	323	206	555	41	17	9	8	654	7918
30-JUL	SUN	3921	1027	528	2326	289	9	399	230	707	54	29	10	3	690	9613
31-JUL	MON	3032	669	496	1496	481	27	365	244	1320	112	17	9	7	730	7866

TRAFFIC VOLUME OF AUGUST 1995, SOUTH SITE

DATE	WEEK	TYPE 1	TYPE 2	TYPE 3	TYPE 4	TYPE 5	TYPE 6	TYPE 7	TYPE 8	TYPE 9	TYPE 10	TYPE 11	TYPE 12	TYPE 13	TYPE 14	TOTAL
1-AUG	TUE	2919	631	447	1329	538	20	381	229	1389	137	25	13	5	654	7468
2-AUG	WED	2892	558	489	1401	601	30	390	253	1500	157	32	13	14	702	7700
3-AUG	THU	3170	622	524	1474	567	43	472	291	1660	180	29	15	14	867	8411
4-AUG	FRI	3809	760	624	1712	461	32	443	296	1351	131	43	14	13	964	9417
5-AUG	SAT	3653	765	662	1677	217	14	377	200	592	34	32	7	8	844	8559
6-AUG	SUN	4363	869	586	2120	284	15	499	256	813	80	30	9	8	864	10052
7-AUG	MON	2756	849	180	990	352	59	195	390	1320	115	25	30	14	710	6924
8-AUG	TUE	2649	861	87	805	248	87	141	392	1391	109	44	35	27	576	6476
9-AUG	WED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10-AUG	THU	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
11-AUG	FRI	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
12-AUG	SAT	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
13-AUG	SUN	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
14-AUG	MON	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
15-AUG	TUE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
16-AUG	WED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
17-AUG	THU	2386	860	303	795	258	23	213	274	1665	97	39	23	31	681	6558
18-AUG	FRI	3427	938	605	1425	399	36	349	236	1311	109	38	12	7	728	8536
19-AUG	SAT	3322	906	528	1378	220	16	306	190	601	38	16	7	0	577	7613
20-AUG	SUN	3806	1168	536	1714	224	7	378	234	756	63	23	10	9	576	8884
21-AUG	MON	2651	729	526	1055	434	33	340	201	1420	102	9	6	15	535	6870
22-AUG	TUE	2816	765	504	1047	471	37	367	221	1434	136	25	10	7	522	7156
23-AUG	WED	2841	771	545	1077	511	48	329	240	1544	117	30	24	16	612	7424
24-AUG	THU	2574	1379	185	631	245	20	154	141	1497	172	41	14	6	244	6748
25-AUG	FRI	3507	2395	47	652	113	1	32	145	1308	29	69	11	0	33	8342
26-AUG	SAT	3663	2184	18	479	48	0	34	73	520	9	48	6	1	32	7114
27-AUG	SUN	4322	2527	14	556	30	0	59	83	796	15	39	4	0	37	8482
28-AUG	M0N	2449	1723	29	486	118	0	30	115	1504	25	61	7	1	38	6586
29-AUG	TUE	2230	1579	39	419	144	0	34	137	1419	15	65	10	0	21	6111
30-AUG	WED	2265	1529	38	477	143	1	23	118	1558	27	57	17	2	42	6297
31-AUG	THU	2553	1788	42	495	155	1	35	117	1609	33	66	10	1	29	6934

TRAFFIC VOLUME OF SEPTEMBER 1995, SOUTH SITE

DATE	WEEK	TYPE 1	TYPE 2	TYPE 3	TYPE 4	TYPE 5	TYPE 6	TYPE 7	TYPE 8	TYPE 9	TYPE 10	TYPE 11	TYPE 12	TYPE 13	TYPE 14	TOTAL
1-SEP	FRI	4527	2573	36	776	139	0	46	131	1301	34	67	10	0	68	9708
2-SEP	SAT	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
3-SEP	SUN	4718	2573	20	559	13	0	37	87	393	4	23	2	0	27	8456
4-SEP	MON	6594	3736	26	978	66	1	95	161	729	7	30	6	0	56	12485
5-SEP	TUE	2939	1901	29	570	95	0	30	91	1401	16	50	5	0	29	7156
6-SEP	WED	2495	1694	37	492	109	0	32	103	1419	35	60	13	0	29	6517
7-SEP	THU	2507	1714	36	540	144	0	20	134	1515	31	64	18	1	41	6765
8-SEP	FRI	3403	2094	39	662	138	1	35	117	1417	24	60	18	0	39	8047
9-SEP	SAT	3513	1979	30	504	43	0	30	90	619	13	51	7	3	26	6908
10-SEP	SUN	3924	2406	24	635	17	0	58	91	790	10	39	5	1	28	8028
11-SEP	MON	2383	1694	35	488	175	1	36	88	1411	20	55	9	1	19	6415
12-SEP	TUE	2265	1592	32	449	116	0	46	102	1489	25	56	11	1	25	6209
13-SEP	WED	2317	1654	29	473	125	1	30	99	1581	32	62	12	0	20	6435
14-SEP	THU	2590	1816	38	484	119	0	31	111	1635	36	70	13	1	34	6978
15-SEP	FRI	3550	2142	46	663	107	1	35	106	1234	14	73	15	1	40	8027
16-SEP	SAT	3588	2083	27	552	39	0	21	98	547	14	48	6	0	17	7040
17-SEP	SUN	4407	2680	23	654	24	1	55	101	840	9	32	6	0	39	8871
18-SEP	MON	2655	1584	33	485	112	1	35	115	1492	21	64	9	2	78	6686
19-SEP	TUE	2777	1203	32	387	82	0	26	116	1394	19	62	9	3	36	6146
20-SEP	WED	2909	1234	38	354	98	2	26	112	1465	19	68	17	0	28	6270
21-SEP	THU	2958	1330	35	411	87	1	22	109	1551	18	81	13	2	28	6646
22-SEP	FRI	4181	1598	36	570	82	0	32	114	1238	24	66	14	1	31	7987
23-SEP	SAT	4079	1521	17	452	28	0	24	76	564	9	50	6	0	19	6845
24-SEP	SUN	5171	2000	14	565	29	0	29	92	799	12	46	5	0	35	8797
25-SEP	MON	3016	1304	39	451	135	0	27	108	1417	9	66	9	0	35	6616
26-SEP	TUE	2696	1265	29	402	162	1	25	99	1447	26	68	16	0	29	6265
27-SEP	WED	2823	1279	26	418	156	0	30	126	1468	25	74	12	1	43	6481
28-SEP	THU	3194	1344	37	431	148	1	30	105	1530	23	65	11	0	17	6936
29-SEP	FRI	4850	1844	37	611	125	0	34	123	1290	28	80	12	0	40	9074
30-SEP	SAT	4411	1721	17	481	34	1	18	68	592	17	43	8	0	16	7427

TRAFFIC VOLUME OF OCTOBER 1995, SOUTH SITE

DATE	WEEK	TYPE 1	TYPE 2	TYPE 3	TYPE 4	TYPE 5	TYPE 6	TYPE 7	TYPE 8	TYPE 9	TYPE 10	TYPE 11	TYPE 12	TYPE 13	TYPE 14	TOTAL
1-OCT	SUN	5430	2353	19	581	14	0	47	108	791	7	42	10	1	37	9440
2-OCT	MON	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
3-OCT	TUE	2850	1321	26	389	103	0	36	117	1450	27	76	17	1	30	6443
4-OCT	WED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
5-OCT	THU	3363	1406	48	443	102	2	33	110	1705	38	78	11	0	24	7363
6-OCT	FRI	4630	1797	37	603	106	0	27	123	1347	22	72	11	2	29	8806
7-OCT	SAT	4508	1617	20	431	38	0	32	77	566	14	55	5	0	10	7373
8-OCT	SUN	5545	2180	23	552	13	0	48	88	845	6	40	5	1	26	9372
9-OCT	MON	3426	1539	28	511	102	2	40	131	1489	17	56	7	1	31	7380
10-OCT	TUE	3011	1373	36	378	101	0	32	112	1377	30	66	12	2	18	6548
11-OCT	WED	2905	1269	27	413	90	0	32	111	1516	21	71	8	0	16	6479
12-OCT	THU	3237	1403	32	439	129	1	35	110	1559	19	69	11	0	38	7082
13-OCT	FRI	4461	1739	38	633	126	1	31	106	1331	12	67	13	0	22	8580
14-OCT	SAT	4293	1695	20	469	37	0	17	70	570	12	49	7	0	10	7249
15-OCT	SUN	5491	2315	13	581	13	0	44	93	767	11	34	7	1	22	9392
16-OCT	MON	3196	1347	36	440	112	0	41	109	1393	15	64	13	0	22	6788
17-OCT	TUE	2862	1305	24	397	139	0	29	105	1422	21	64	14	2	21	6405
18-OCT	WED	2935	1322	35	436	134	2	30	132	1550	26	56	11	2	33	6704
19-OCT	THU	3291	1415	38	451	128	1	43	123	1607	24	74	13	2	25	7235
20-OCT	FRI	4653	1758	25	578	111	0	25	116	1231	18	67	16	0	33	8631
21-OCT	SAT	4394	1761	18	485	46	0	20	94	564	11	48	8	0	20	7469
22-OCT	SUN	5651	2262	18	685	20	0	48	111	779	8	41	7	0	37	9667
23-OCT	MON	3075	1446	30	429	109	1	31	115	1550	22	55	7	1	32	6903
24-OCT	TUE	2953	1320	31	418	109	0	31	113	1433	19	61	15	1	25	6529
25-OCT	WED	2929	1289	20	436	114	2	27	104	1499	18	63	12	1	23	6537
26-OCT	THU	3134	1388	35	469	106	0	29	135	1569	20	64	20	1	35	7004
27-OCT	FRI	4353	1744	42	602	104	1	40	114	1286	24	72	15	0	27	8424
28-OCT	SAT	4523	1747	24	470	31	0	18	94	577	16	54	11	1	16	7582
29-OCT	SUN	5836	2302	16	630	24	0	40	93	724	11	34	7	0	34	9751
30-OCT	MON	3108	1442	32	463	127	0	29	111	1440	8	54	12	1	29	6856
31-OCT	TUE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

TRAFFIC VOLUME OF NOVEMBER 1995, SOUTH SITE

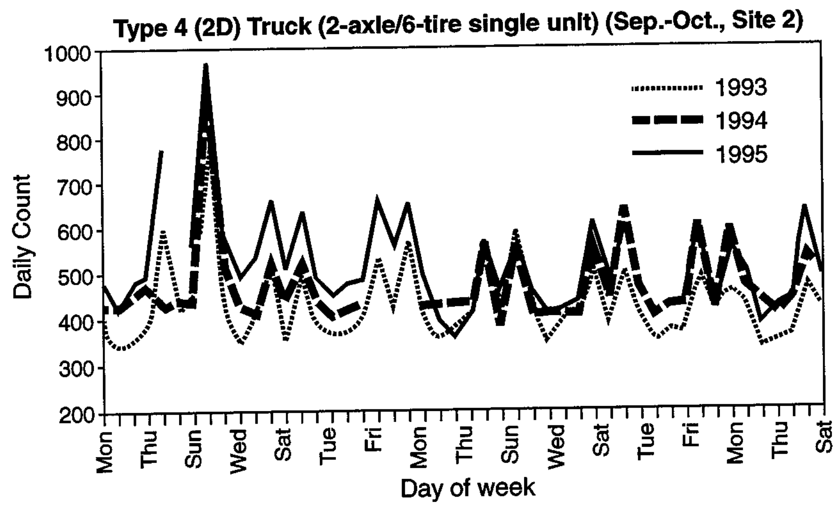
DATE	WEEK	TYPE 1	TYPE 2	TYPE 3	TYPE 4	TYPE 5	TYPE 6	TYPE 7	TYPE 8	TYPE 9	TYPE 10	TYPE 11	TYPE 12	TYPE 13	TYPE 14	TOTAL
1-NOV	WED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2-NOV	THU	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
3-NOV	FRI	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
4-NOV	SAT	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
5-NOV	SUN	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
6-NOV	MON	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
7-NOV	TUE	2848	1434	34	393	79	0	29	111	1288	23	45	15	1	17	6317
8-NOV	WED	2942	1361	29	419	96	0	22	96	1373	27	64	14	0	26	6469
9-NOV	THU	3295	1453	38	456	96	2	31	97	1591	10	69	18	0	25	7181
10-NOV	FRI	4194	1717	26	604	132	0	37	103	1195	22	68	15	0	19	8132
11-NOV	SAT	4385	1644	24	423	31	0	18	60	535	11	48	8	0	20	7207
12-NOV	SUN	5229	2349	22	590	15	0	37	91	749	4	34	7	1	24	9151
13-NOV	MON	3042	1378	22	411	84	1	34	95	1432	17	59	10	0	18	6603
14-NOV	TUE	2977	1357	31	396	88	0	33	96	1400	20	53	12	0	20	6483
15-NOV	WED	2938	1313	42	404	121	0	20	103	1489	20	60	13	1	34	6558
16-NOV	THU	3195	1441	40	404	108	0	26	106	1645	22	60	9	0	26	7082
17-NOV	FRI	4319	1697	47	518	125	0	27	102	1350	24	74	13	2	24	8322
18-NOV	SAT	4408	1674	30	488	37	0	21	79	573	7	45	12	0	10	7384
19-NOV	SUN	4792	2050	16	535	13	0	29	99	862	8	30	2	0	29	8465
20-NOV	MON	3223	1317	25	459	90	0	28	108	1528	12	59	9	0	20	6878
21-NOV	TUE	4215	1640	41	545	109	1	18	94	1489	12	59	14	2	32	8271
22-NOV	WED	7355	2343	39	863	100	1	48	134	1275	14	54	16	0	45	12287
23-NOV	THU	5561	1751	12	455	14	0	19	51	399	2	24	7	0	14	8309
24-NOV	FRI	5980	2017	21	578	34	0	23	74	556	5	7	2	0	21	9318
25-NOV	SAT	8315	2892	18	791	18	0	30	126	311	5	27	1	0	32	12566
26-NOV	SUN	10564	3988	23	1122	17	0	48	178	632	6	38	9	0	60	16685
27-NOV	MON	3748	1530	23	476	102	1	28	105	1451	17	64	13	0	34	7592
28-NOV	TUE	2975	1311	31	395	104	0	28	104	1444	13	55	15	0	17	6492
29-NOV	WED	2809	1294	28	365	82	0	19	124	1461	20	57	17	1	27	6304
30-NOV	THU	3028	1347	45	383	111	0	33	105	1555	24	74	14	0	15	6734

TRAFFIC VOLUME OF DECEMBER 1995, SOUTH SITE

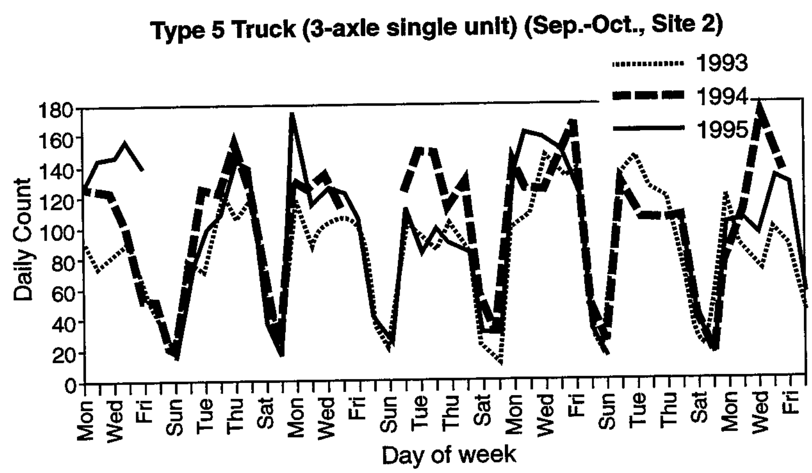
DATE	WEEK	TYPE 1	TYPE 2	TYPE 3	TYPE 4	TYPE 5	TYPE 6	TYPE 7	TYPE 8	TYPE 9	TYPE 10	TYPE 11	TYPE 12	TYPE 13	TYPE 14	TOTAL
1-DEC	FRI	4018	1596	38	515	119	0	31	98	1260	14	60	19	0	16	7774
2-DEC	SAT	3832	1552	27	425	45	0	17	76	556	12	58	12	1	22	6635
3-DEC	SUN	4695	1967	23	545	20	0	23	89	816	5	38	5	0	22	8248
4-DEC	MON	3085	1358	34	403	127	0	22	111	1500	22	53	14	0	17	6746
5-DEC	TUE	2813	1308	40	419	129	0	25	114	1437	25	62	15	1	23	6411
6-DEC	WED	2795	1268	34	461	121	0	20	120	1539	19	70	18	0	18	6483
7-DEC	THU	3071	1354	47	391	99	0	34	133	1561	14	67	17	1	24	6803
8-DEC	FRI	3714	1547	33	465	66	1	29	108	1203	31	60	15	1	27	7300
9-DEC	SAT	3808	1454	20	386	24	0	7	71	507	12	53	10	0	12	6364
10-DEC	SUN	4102	1807	14	400	13	0	17	65	752	8	34	2	1	13	7228
11-DEC	MON	2905	1261	26	402	94	1	18	87	1437	31	47	4	0	18	6331
12-DEC	TUE	2818	1207	35	380	77	1	24	106	1362	25	58	13	0	18	6124
13-DEC	WED	3007	1340	31	424	69	1	25	103	1471	26	64	14	0	14	6589
14-DEC	THU	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
15-DEC	SAT	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
16-DEC	FRI	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
17-DEC	SUN	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
18-DEC	MON	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
19-DEC	TUE	3563	1438	20	439	79	0	30	89	1267	15	65	8	2	21	7036
20-DEC	WED	3786	1532	29	435	77	1	21	100	1421	34	57	8	1	22	7524
21-DEC	THU	4409	1717	36	481	81	0	24	90	1481	28	63	15	0	33	8458
22-DEC	FRI	5497	1894	23	544	68	1	30	108	1027	16	64	10	0	18	9300
23-DEC	SAT	6729	2101	15	565	19	0	13	75	451	7	36	1	0	26	10037
24-DEC	SUN	5600	1769	8	435	8	0	14	47	187	0	1	2	0	9	8080
25-DEC	MON	6536	1846	11	559	8	0	14	68	179	2	1	0	0	17	9241
26-DEC	TUE	7661	2459	25	753	50	0	43	125	829	15	35	5	1	38	12039
27-DEC	WED	5341	1832	17	561	44	1	23	131	1169	15	56	8	1	24	9223
28-DEC	THU	4811	1794	22	529	73	0	37	133	1285	18	73	11	3	27	8816
29-DEC	FRI	5289	1880	19	548	114	0	32	134	1055	15	54	10	0	38	9188
30-DEC	SAT	5697	1955	14	576	21	0	21	87	509	4	54	6	0	14	8958
31-DEC	SUN	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N

APPENDIX C:

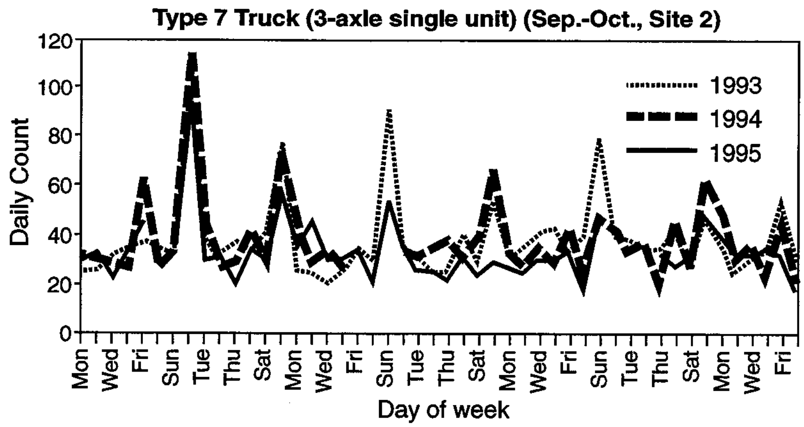
TRAFFIC GROWTH RATE ANALYSIS

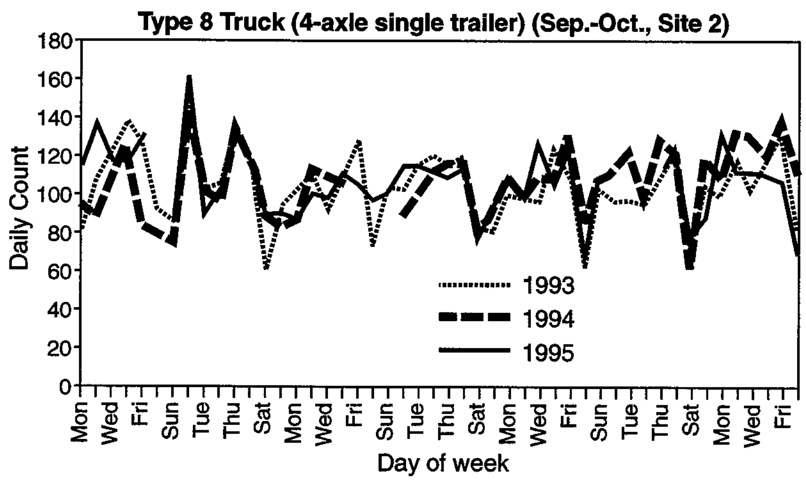


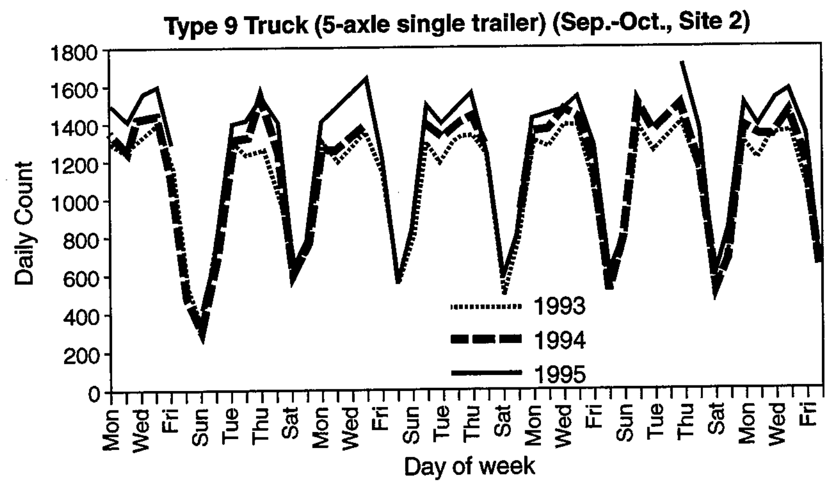
Type 4 (2D) Truck (2-axle/6-tire single unit) (Sep.-Oct., Site 2)
1993
1994
1995
Daily Count
Day of week

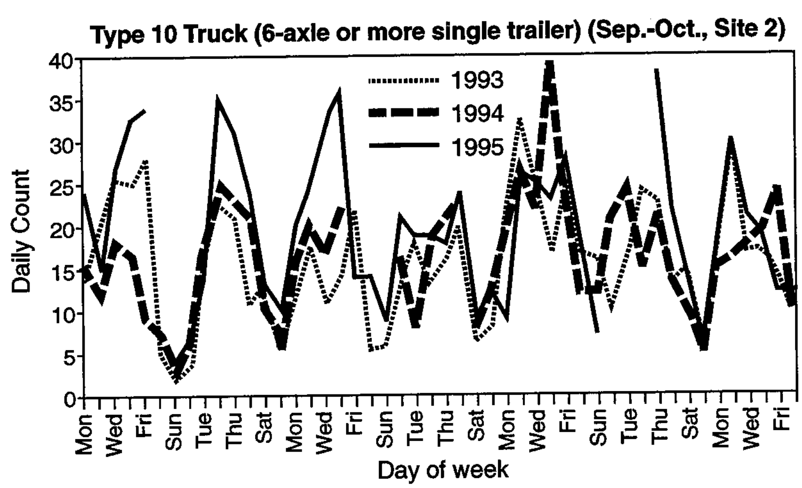


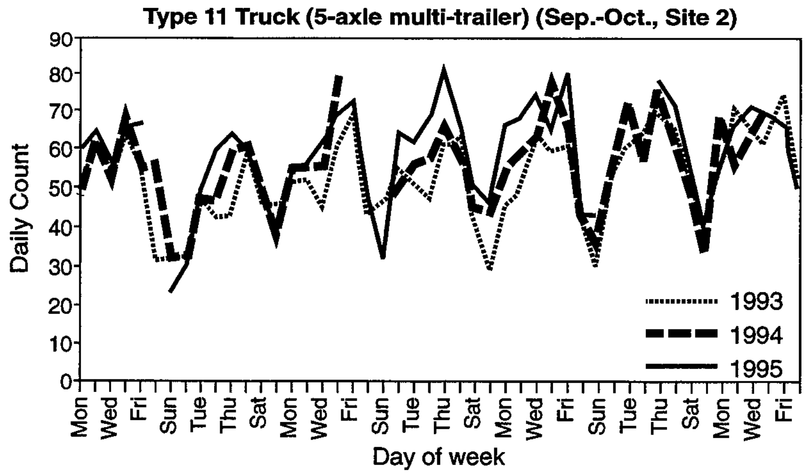
Type 5 Truck (3-axle single unit) (Sep.-Oct., Site 2)
1993
1994
1995
Daily Count
Day of week

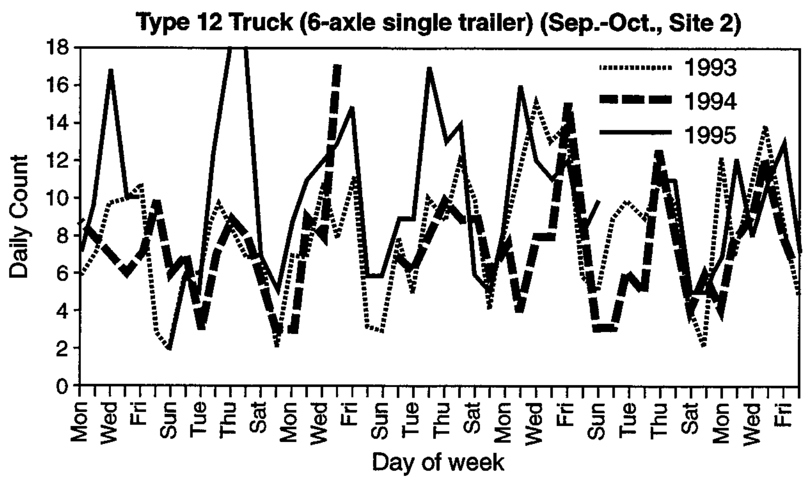

Type 7 Truck (3-axle single unit) (Sep.-Oct., Site 2)
Daily Count
0
20
40
60
80
100
120
1993
1994
1995
Mon
Wed
Fri
Sun
Tue
Thu
Sat
Mon
Wed
Fri
Sun
Tue
Thu
Sat
Mon
Wed
Fri
Sun
Tue
Thu
Sat
Mon
Wed
Fri
Day of week


Type 8 Truck (4-axle single trailer) (Sep.-Oct., Site 2)
Daily Count
0
20
40
60
80
100
120
140
160
180
1993
1994
1995
Mon
Wed
Fri
Sun
Tue
Thu
Sat
Mon
Wed
Fri
Sun
Tue
Thu
Sat
Mon
Wed
Fri
Sun
Tue
Thu
Sat
Mon
Wed
Fri
Day of week


Type 9 Truck (5-axle single trailer) (Sep.-Oct., Site 2)
Daily Count
0
200
400
600
800
1000
1200
1400
1600
1800
1993
1994
1995
Mon
Wed
Fri
Sun
Tue
Thu
Sat
Day of week


Type 10 Truck (6-axle or more single trailer) (Sep.-Oct., Site 2)
Daily Count
0
5
10
15
20
25
30
35
40
1993
1994
1995
Mon
Wed
Fri
Sun
Tue
Thu
Sat
Day of week


Type 11 Truck (5-axle multi-trailer) (Sep.-Oct., Site 2)
Daily Count
Day of week
1993
1994
1995
90
80
70
60
50
40
30
20
10
0
Mon
Wed
Fri
Sun
Tue
Thu
Sat


Type 12 Truck (6-axle single trailer) (Sep.-Oct., Site 2)
Daily Count
Day of week
1993
1994
1995
18
16
14
12
10
8
6
4
2
0
Mon
Wed
Fri
Sun
Tue
Thu
Sat

APPENDIX D:

INSTRUCTIONS FOR USING CAR AND ESAL PROGRAM

This appendix is included to provide guidance for using the computer programs developed for this report. There are two PC-format programs incorporated into this study. The first one is used for sorting vehicles by TxDOT classification and for calculating the axle-load frequency distribution and the weighted average ESALs for each type of TxDOT-classified vehicles. The second program is for forecasting the ESALs on a user-defined pavement type for a user-defined period.

The executive file of the first program is named "CAR" and the executive file of the second program is called "ESAL." The source files and their respective computer-occupied spaces of the two programs are listed in Table D-1:

Table D-1 Name and space of the programs

	CAR	ESAL
Name of Executive file:	car.exe	esal.exe
# of bytes:	143,484	56,768
Name of source file:	tc2.cpp	esal.cpp
# of bytes:	34,686	3,016

It should be noted that users need only the executive file, such as car.exe, to execute the program without using the resource file, such as, tc2.cpp; the executive files can be executed without any prerequested background. For example, the program can be executed by entering "car" or "esal" under the DOS command or by clicking the executive files in the File Manager of Windows' background. The executive commands are not case sensitive.

It is suggested by the author that users set up a new directory for the use of the programs. Users should first copy both car.exe and esal.exe into the created directory, and then copy the data files that are to be analyzed.

D.1 CAR PROGRAM -FUNCTION 1: SORTING VEHICLES

When executing the CAR program, users will be prompted to input a couple of parameters corresponding to the questions popped up consecutively on the screen (see Figure D-1). The first three parameters are about the data files that users want to analyze. Users need to specify the year, month, and site of the data files. If users want to analyze a whole year's data at one time without inputting the parameters for each month, users should input "*" for the month parameter. Users also need to name the output file and choose between "count" and "weight" functions. For example, if a user wants to sort vehicles, then the user needs to choose the first option: 1. After the user keys in "1" and hits the enter key, the program will start to count and sort the vehicles, and the user will see the hundred numbers showing up on the screen. Those cumulative numbers tell the user the number of vehicles sorted by the program. After the program finishes one day's data, the number will start from 100 again for the next day's data. For example, if there were more than 6400 but fewer than 6500 vehicles in the first day's data, after the number 6400, the number 100 will be shown on the screen again, which means the program has already started to sort the second day's data.

Once the program has finished all the data files in the user-defined month, the lower left corner of the screen will change from "Running" to "Finished." It takes about 10 minutes to sort one month's data. A sample screen for the sorting vehicles function (option 1) of the CAR program is shown in Figure D-1 and the output file c1 of March 23 of 1994's data is shown in Table D-2. The output file can be opened from DOS, Microsoft EXCEL, and from Word.

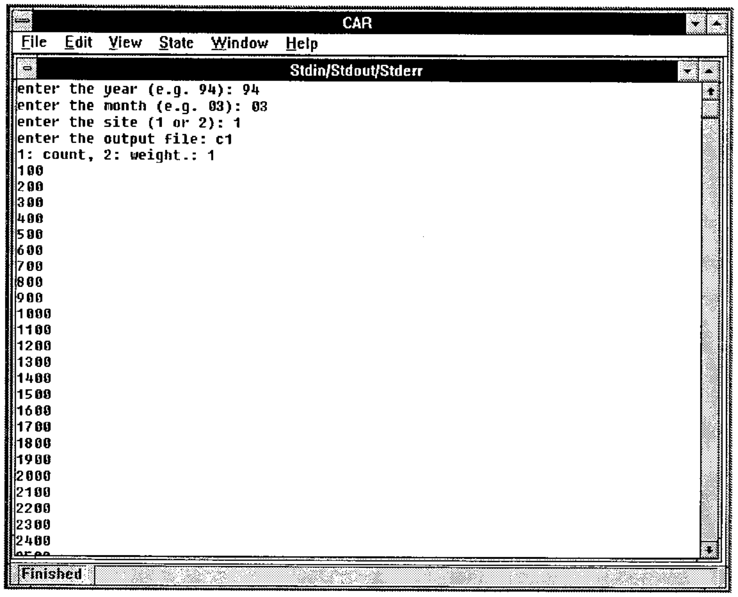


Figure D-1 Screen for the vehicle-sorting function (option 1) of CAR program

Table D-2 Output file format for vehicle-sorting function of CAR program

Traffic type	count on lane 1	23-Mar lane 2
1	1967	650
2	1202	357
3	26	5
4	272	176
5	59	12
6	1	0
7	34	5
8	74	12
9	1289	159
10	20	0
11	54	7
12	10	0
13	1	0
14	8	3
Total	5017	1386

D.2 CAR PROGRAM — FUNCTION 2: AXLE WEIGHT FREQUENCY DISTRIBUTION AND WEIGHTED AVERAGE ESALS FOR EACH VEHICLE TYPE

When the user chooses the weight function (option 2) in the fifth question of the CAR program, the program prompts for more user input (see Figure D-2). The user needs to specify pavement type with its terminal serviceability and structural number (flexible pavement) or slab thickness (rigid pavement). The structural numbers range from 1 to 9 for flexible pavement, and from 6 to 14 for rigid pavement.

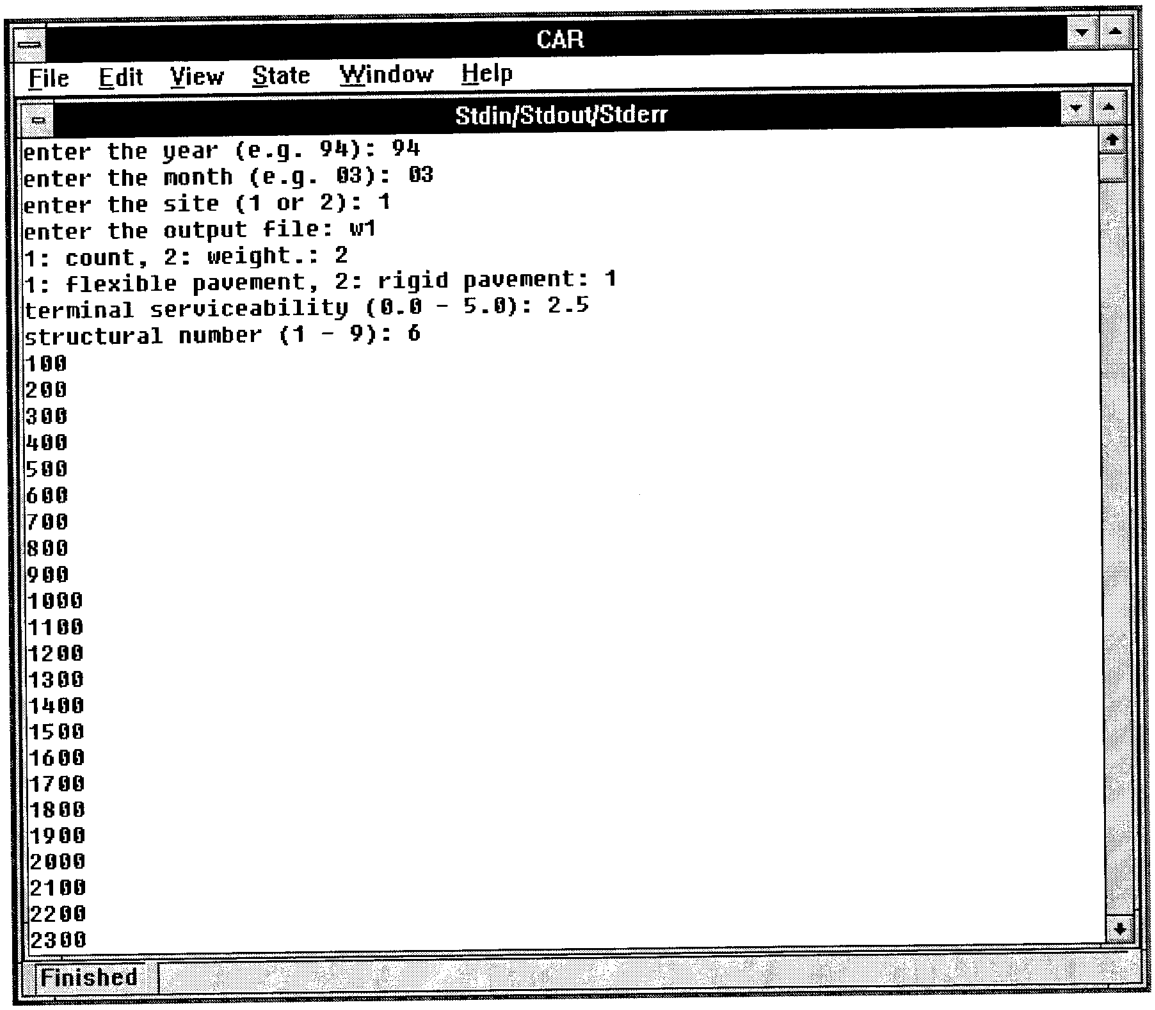


Figure D-2 Screen for axle-load distribution function (option 2) of CAR program

The output file w1 for the weight distribution function of March 23 of 1995's data is shown in Table D-3.

Table D-3 Output file w1 for March 23, 1995, by axle-load distribution function of CAR program

TYPE 4:

	S41		S42		S43		S44	
weight range	# of axles	weight distri.	# of axles	weight distri.	# of axles	weight distri.	# of axles	weight distri.
2	214	53.9	4	20	13	68.42	0	0
4	55	13.85	2	10	1	5.26	10	32.26
6	43	10.83	4	20	2	10.53	17	54.84
8	28	7.05	4	20	1	5.26	1	3.23
10	24	6.05	1	5	1	5.26	3	9.68
12	7	1.76	0	0	1	5.26	0	0
14	7	1.76	1	5	0	0	0	0
16	5	1.26	0	0	0	0	0	0
18	7	1.76	1	5	0	0	0	0
20	3	0.76	1	5	0	0	0	0
22	2	0.5	2	10	0	0	0	0
24	2	0.5	0	0	0	0	0	0
total	397		20		19		31	

	T4	
weight range	# of axles	weight distri.
4	23	74.19
8	8	25.81
12	0	0
16	0	0
20	0	0
24	0	0
28	0	0
32	0	0
36	0	0
40	0	0
44	0	0
48	0	0
total	31	

weight range	E41 # of axles	weight distri.	E42 # of axles	weight distri.	E43 # of axles	weight distri.
13	2	100	0	0	0	0
15	0	0	1	100	0	0
17	0	0	0	0	0	0
19	0	0	0	0	0	0
total	2		1		0	

TYPE 5:

weight range	T5 # of axles	weight distri.
4	0	0
8	13	18.31
12	11	15.49
16	14	19.72
20	15	21.13
24	3	4.23
28	4	5.63
32	4	5.63
36	2	2.82
40	3	4.23
44	2	2.82
48	0	0
total	71	

weight range	E5 # of axles	weight distri.
13	1	1
15	0	0
17	0	0
19	0	0
total	1	

TYPE 6:

	S6	
weight range	# of axles	weight distri.
2	0	0
4	0	0
6	0	0
8	0	0
10	0	0
12	0	0
14	0	0
16	1	100
18	0	0
20	0	0
22	0	0
24	0	0
total	1	

	T6	
weight range	# of axles	weight distri.
4	0	0
8	0	0
12	0	0
16	0	0
20	0	0
24	0	0
28	0	0
32	1	100
36	0	0
40	0	0
44	0	0
48	0	0
total	1	

	E61		E62	
weight range	# of axles	weight distri.	# of axles	weight distri.
13	0	0	0	0
15	0	0	0	0
17	1	1	0	0
19	0	0	0	0
total	1		0	

weight range	R6 # of axles	weight distri.
4	0	0
8	0	0
12	0	0
16	0	0
20	0	0
24	0	0
28	0	0
32	0	0
36	0	0
40	0	0
44	0	0
48	0	0
52	0	0
56	0	0
60	0	0
64	0	0
68	0	0
total	0	

TYPE 7:

weight range	S71 # of axles	weight distri.	S72 # of axles	weight distri.
2	12	30.77	18	46.15
4	3	7.69	2	5.13
6	1	2.56	0	0
8	1	2.56	7	17.95
10	4	10.26	2	5.13
12	5	12.82	2	5.13
14	5	12.82	1	2.56
16	2	5.13	7	17.95
18	6	15.38	0	0
20	0	0	0	0
22	0	0	0	0
24	0	0	0	0
total	39		39	

	E7	
weight range	# of axles	weight distri.
13	0	0
15	1	0.33
17	0	0
19	2	0.67
total	3	

TYPE 8:

	S81		S82	
weight range	# of axles	weight distri.	# of axles	weight distri.
2	0	0	26	54.17
4	2	5.26	1	2.08
6	7	18.42	3	6.25
8	3	7.89	5	10.42
10	7	18.42	6	12.5
12	9	23.68	2	4.17
14	4	10.53	1	2.08
16	2	5.26	3	6.25
18	1	2.63	0	0
20	3	7.89	0	0
22	0	0	1	2.08
24	0	0	0	0
total	38		48	

	T81		T82	
weight range	# of axles	weight distri.	# of axles	weight distri.
4	3	7.5	13	28.26
8	14	35	6	13.04
12	9	22.5	8	17.39
16	9	22.5	6	13.04
20	3	7.5	6	13.04
24	1	2.5	1	2.17
28	1	2.5	0	0
32	0	0	1	2.17
36	0	0	2	4.35
40	0	0	3	6.52
44	0	0	0	0
48	0	0	0	0
total	40		46	

	E81		E82	
weight range	# of axles	weight distri.	# of axles	weight distri.
13	0	0	2	1
15	0	0	0	0
17	0	0	0	0
19	0	0	0	0
total	0		2	

TYPE 9:

	S9		DS91		DS92	
weight range	# of axles	weight distri.	# of axles	weight distri.	# of axles	weight distri.
2	0	0	0	0	1	0.85
4	0	0	3	2.54	4	3.39
6	3	42.86	7	5.93	7	5.93
8	3	42.86	6	5.08	5	4.24
10	1	14.29	10	8.47	13	11.02
12	0	0	6	5.08	9	7.63
14	0	0	17	14.41	8	6.78
16	0	0	33	27.97	36	30.51
18	0	0	27	22.88	28	23.73
20	0	0	6	5.08	6	5.08
22	0	0	1	0.85	1	0.85
24	0	0	0	0	0	0
total	7		118		118	

	T91		T92		DT9	
weight range	# of axles	weight distri.	# of axles	weight distri.	# of axles	weight distri.
4	3	0.23	16	1.21	1	0.85
8	19	1.44	225	16.96	3	2.54
12	242	18.33	153	11.53	3	2.54
16	132	10	104	7.84	10	8.47
20	126	9.55	106	7.99	7	5.93
24	120	9.09	116	8.74	15	12.71
28	153	11.59	216	16.28	36	30.51
32	333	25.23	324	24.42	38	32.2
36	151	11.44	55	4.14	5	4.24
40	34	2.58	11	0.83	0	0
44	7	0.53	1	0.08	0	0
48	0	0	0	0	0	0
total	1320		1327		118	

	E91		E92		E93	
weight range	# of axles	weight distri.	# of axles	weight distri.	# of axles	weight distri.
13	0	0	61	0.94	3	1
15	0	0	4	0.06	0	0
17	0	0	0	0	0	0
19	0	0	0	0	0	0
total	0		65		3	

	R9	
weight range	# of axles	weight distri.
4	0	0
8	3	42.86
12	2	28.57
16	2	28.57
20	0	0
24	0	0
28	0	0
32	0	0
36	0	0
40	0	0
44	0	0
48	0	0
52	0	0
56	0	0
60	0	0
64	0	0
68	0	0
total	7	

TYPE 10 :

	T10	
weight range	# of axles	weight distri.
4	0	0
8	0	0
12	0	0
16	2	10
20	6	30
24	4	20
28	5	25
32	2	10
36	0	0
40	1	5
44	0	0
48	0	0
total	20	

weight range	E10 # of axles	weight distri.
13	1	0.5
15	0	0
17	1	0.5
19	0	0
total	2	

weight range	R10 # of axles	weight distri.
4	0	0
8	0	0
12	7	35
16	3	15
20	0	0
24	3	15
28	1	5
32	1	5
36	0	0
40	3	15
44	0	0
48	1	5
52	1	5
56	0	0
60	0	0
64	0	0
68	0	0
total	20	

TYPE 11 :

weight range	S111 # of axles	weight distri.	S112 # of axles	weight distri.	S113 # of axles	weight distri.	S114 # of axles	weight distri.
2	1	1.64	4	6.56	4	6.56	7	11.48
4	2	3.28	0	0	0	0	1	1.64
6	0	0	2	3.28	6	9.84	4	6.56
8	1	1.64	1	1.64	7	11.48	12	19.67
10	1	1.64	4	6.56	10	16.39	5	8.2
12	8	13.11	11	18.03	15	24.59	12	19.67
14	7	11.48	9	14.75	8	13.11	6	9.84
16	16	26.23	13	21.31	7	11.48	8	13.11
18	16	26.23	12	19.67	3	4.92	4	6.56
20	6	9.84	5	8.2	1	1.64	2	3.28
22	3	4.92	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0
total	61		61		61		61	

	E11	
weight range	# of axles	weight distri.
13	2	0.5
15	1	0.25
17	1	0.25
19	0	0
total	4	

TYPE 12 :

	S1211		S1212		S1213	
weight range	# of axles	weight distri.	# of axles	weight distri.	# of axles	weight distri.
2	0	0	0	0	0	0
4	0	0	0	0	0	0
6	0	0	0	0	0	0
8	0	0	0	0	0	0
10	0	0	0	0	0	0
12	0	0	0	0	0	0
14	0	0	0	0	0	0
16	0	0	0	0	0	0
18	0	0	0	0	0	0
20	0	0	0	0	0	0
22	0	0	0	0	0	0
24	0	0	0	0	0	0
total	0		0		0	

	S1221		S1222		S1223	
weight range	# of axles	weight distri.	# of axles	weight distri.	# of axles	weight distri.
2	0	0	0	0	0	0
4	0	0	0	0	0	0
6	1	10	3	30	3	30
8	0	0	0	0	1	10
10	0	0	3	30	3	30
12	2	20	1	10	2	20
14	5	50	2	20	0	0
16	0	0	0	0	0	0
18	2	20	0	0	1	10
20	0	0	1	10	0	0
22	0	0	0	0	0	0
24	0	0	0	0	0	0
total	10		10		10	

	T121		T122	
weight range	# of axles	weight distri.	# of axles	weight distri.
4	0	0	0	0
8	0	0	0	0
12	0	0	1	10
16	0	0	4	40
20	0	0	5	50
24	0	0	0	0
28	0	0	0	0
32	0	0	0	0
36	0	0	0	0
40	0	0	0	0
44	0	0	0	0
48	0	0	0	0
total	0		10	

	E121		E122	
weight range	# of axles	weight distri.	# of axles	weight distri.
13	0	0	0	0
15	0	0	0	0
17	0	0	0	0
19	0	0	0	0
total	0		0	

TYPE 13 :

	S131		S132	
weight range	# of axles	weight distri.	# of axles	weight distri.
2	1	100	1	100
4	0	0	0	0
6	0	0	0	0
8	0	0	0	0
10	0	0	0	0
12	0	0	0	0
14	0	0	0	0
16	0	0	0	0
18	0	0	0	0
20	0	0	0	0
22	0	0	0	0
24	0	0	0	0
total	1		1	

	T131		T132	
weight range	# of axles	weight distri.	# of axles	weight distri.
4	0	0	0	0
8	0	0	0	0
12	0	0	1	100
16	0	0	0	0
20	1	100	0	0
24	0	0	0	0
28	0	0	0	0
32	0	0	0	0
36	0	0	0	0
40	0	0	0	0
44	0	0	0	0
48	0	0	0	0
total	1		1	

weight range	E13 # of axles	weight distri.
13	0	0
15	0	0
17	0	0
19	0	0
total	0	

------- ------ --------- ------- --------- ------

Type 4
2D : 0 0.08257
2D-1 : 0.403
2D-2 : 0.015

Type 5
3A : 0 0.33583

Type 6
4A : 0 0
Rig : 1.4401

Type 7
2S1 : 0.3972

Type 8
2S2 : 0.317
3S1 : 0.3592

Type 9
2S3 : 0.0326
3S3 : 0.9253
3S2 : 1.6437

Type 10
3S3 : 0.4997

Type 11
2S1-2 : 1.82

Type 12
2S2-2 : 0.00
3S1-2 : 0.91

Type 13
3S2-2 : 0.12

D.3 ESAL PROGRAM

The ESAL program is used for forecasting the cumulative ESALs at the end of the selected forecasting period. When executing the ESAL program, the user must input the forecasting period, either from simple or compound growth rate (see Section 5.3), the growth rate, the AADT, as well as the weighted average ESALs for each type of vehicle (see Figure D-3).

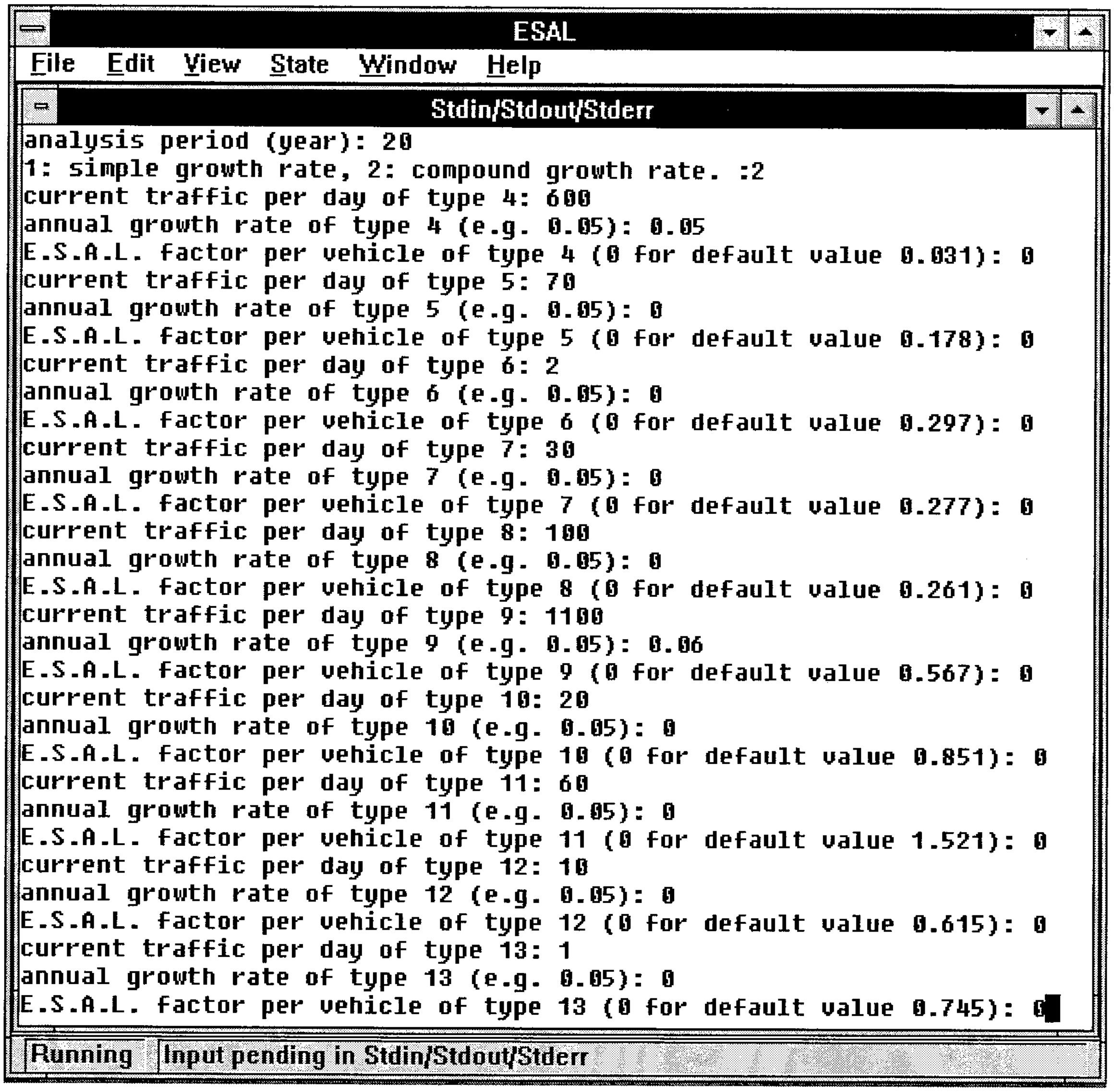


Figure D-3 Screen for ESAL program (part 1)

After the user inputs the AADT, growth rate, and weighted average ESALs for each type of vehicle, the program will list all the data again for the user to check. For example, if the user wants to modify the growth rate for the type 9 vehicle, the user needs to retype the three parameters for the type 9 vehicle (see Figure D-4). After the user has modified the selected type of vehicle, the program will still provide opportunities for modification until the user is satisfied with the inputs. When the user inputs 0, the program will start to calculate; it takes only about 2 seconds for the program to calculate the total cumulative ESALs. A sample output file of the ESAL program is shown in Table D-4.

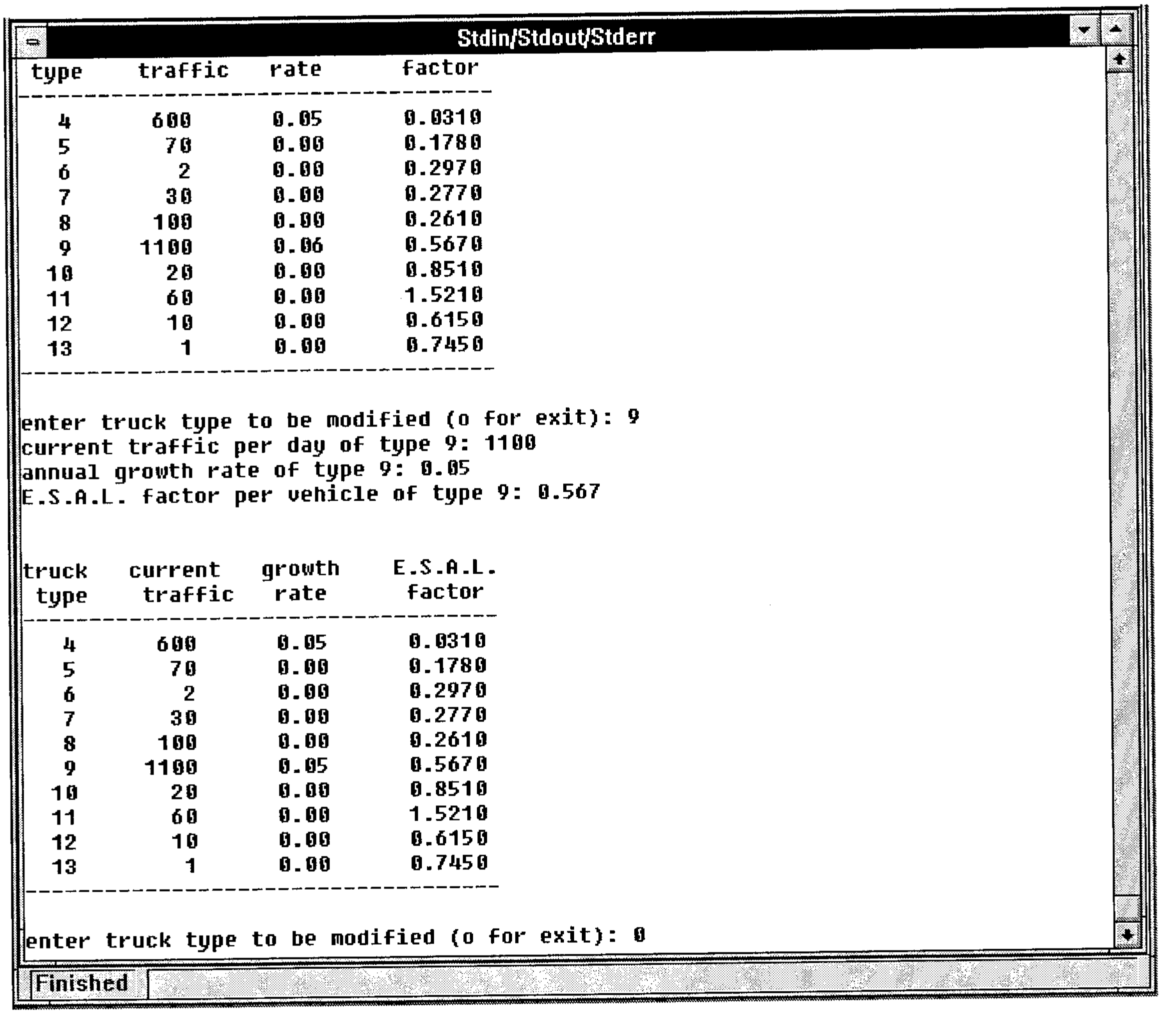


Figure D-4 Screen for ESAL program (part 2)

Table D-4 Output file format for ESAL program

truck types	current traffic (AADT)	growth rate	design traffic (cumulation)	E.S.A.L. factor (/veh.)	design E.S.A.L.
-----	----------	--------	--------------	-----------	------------
4	600	5%	7241235	0.031	224478
5	70	0%	511000	0.178	90957
6	2	0%	14600	0.297	4336
7	30	0%	219000	0.277	60663
8	100	0%	730000	0.261	190530
9	1100	5%	14769360	0.567	8374227
10	20	0%	146000	0.851	124245
11	60	0%	438000	1.521	666197
12	10	0%	73000	0.615	44894
13	1	0%	7300	0.745	5438
-----	----------	--------	--------------	-----------	------------
total	1993		24149495		9785965

Center for Transportation Research
The University of Texas at Austin
3208 Red River, Suite 200
Austin, TX 78705

(512) 232-3100
FAX: (512) 232-3153
Email: transres@www.utexas.edu
Internet: http://www.utexas.edu/depts/ctr